Mikko Koski

Effective communication media for customer feedback in agile software projects

School of Science

Thesis submitted for examination for the degree of Master of Science in Technology.
Espoo 8.5.2013

Thesis supervisor:

Prof. Tapio Takala

Thesis advisor:

Doc. Risto Sarvas, D.Sc.(Tech.)

AALTO-YLIOPISTO
PERUSTIETEIDEN KORKEAKOULU

DIPLOMITYÖN
TIIVISTELMÄ

Tekijä: Mikko Koski

Työn nimi: Tehokkaat viestintävälineet asiakaspalautteelle ketterissä ohjelmistoprojekteissa

Päivämäärä: 8.5.2013 Kieli: Englanti Sivumäärä:8+74

Mediatekniikan laitos

Professuuri: Mediatekniikka Koodi: T-111

Valvoja: Prof. Tapio Takala

Ohjaaja: Dos. Risto Sarvas, TkT

Ohjelmistokehitysmenetelmät ovat siirtyneet vesiputousmallista kohti iteratiivista ja ketterää kehitystä. Ketterässä kehityksessä ohjelmistoa ei rakenneta yhtenä isona työvaiheena, vaan lyhyissä iteraatioissa, joissa jokaisessa tuotetaan pieni lisäys ohjelmistoon. Tämä avaa uusia mahdollisuuksia ja vastuita ohjelmistoprojektin asiakkaalle, jolla on vastuu ohjailla projektia palautteen avulla.

Useat tutkimukset ovat tutkineet kommunikaatiota ketterissä ohjelmistoprojekteissa, mutta vain harva tutkimus on keskittynyt palautekommunikaatioon siitä huolimatta, että sen on tunnistettu olevan yksi tärkeimmistä elementeistä onnistuneissa ohjelmistoprojekteissa. Tässä diplomityössä keskitytään palautekommunikaatioon ja yritetään vastata tutkimuskysymykseen *mitkä ominaisuudet tekevät viestintävälineestä tehokkaan palautteen antamiseen ja saamiseen ketterissä ohjelmistoprojekteissa?*

Tässä työssä käytetään neljää kommunikaatiomediateoriaa vastaamaan tutkimuskysymykseen. Nämä teoriat ovat Media Richness, Media Synchronicity, Media Naturalness ja Media Fitness. Teorioista muodostetaan hypoteesi ominaisuuksista, jotka ovat arvokkaita palautetyökalulle. Hypoteesi validoidaan haastattelemalla Hannotaatio -prototyypin käyttäjiä. Hannotaatio on työkalu visuaalisen palautteen antamiseen web-sivustosta ottamalla kuvakaappaus sivusta ja piirtämällä sen päälle. Prototyypin toimivuutta validoidaan haastatteluilla.

Työn tuloksena huomattiin, että siitä huolimatta että haastateltavat arvostivat kasvokkain kommunikointia, he arvostivat myös sellaisia viestintävälineen ominaisuuksia, joita kasvokkain keskustelussa ei ole. Hannotaatio pyrkii kasvattamaan keskustelun luonnollisuutta ja kaventamaan eroja kasvokkain keskustelun ja elektronisten kommunikaatiovälineiden välillä. Haastateltavat totesivat, että Hannotaatio on toiseksi paras vaihtoehto palautevälineeksi kasvokkain keskustelun jälkeen.

Avainsanat: kommunikaatio, viestintävälineet, ketterä ohjelmistotuotanto, asiakaspalaute

AALTO UNIVERSITY
SCHOOL OF SCIENCE

ABSTRACT OF THE
MASTER'S THESIS

Author: Mikko Koski

Title: Effective communication media for customer feedback in agile software projects

Date: 8.5.2013 Language: English Number of pages:8+74

Department of Media Technology

Professorship: Media Technology Code: T-111

Supervisor: Prof. Tapio Takala

Advisor: Doc. Risto Sarvas, D.Sc.(Tech.)

Software development methodologies have shifted from waterfall development to iterative agile development. Instead of building an application in one big piece, iterative development relies on small and frequently delivered increments. This opens new possibilities and responsibilities for a software project's customer to provide feedback for the develoment team in order to guide the project.

Various studies about communication in agile software projects have been conducted but not so many with a clear focus on feedback communication, even though it has been identified to be one of the key elements in a successful agile project. In this thesis, feedback communication is taken into a focus by trying to answer the research question *what are the properties that make a communication tool effective for giving and receiving feedback in agile software projects?*

Four communication media theories have been applied in this thesis to answer the research question. The theories are media richness, media synchronicity, media naturalness and media fitness theory. Based on these theories, a hypothesis is constructed about the properties that are valuable for a feedback tool.

To validate the hypothesis a prototype called Hannotaatio has been built. Hannotaatio is a tool for giving visual feedback about websites by taking a screenshot and drawing on them. The prototype is validated by interviewing people who have used it in real-life software projects.

The result of the study was that even though interviewees valued face-to-face communication and argued it to be the most efficient communication medium, they also valued properties in communication media that are not available in face-to-face communication. Hannotaatio tries to increase the naturalness in the communication and thus shortens the gap between face-to-face communication and e-communication media. Interviewees noted that Hannotaatio is the second best option for feedback communication after face-to-face communication.

Keywords: communication, agile, software, feedback, media

Preface

Software has been close to my heart my whole life. When I was a young kid, I jumped into the programming world by typing a small piece of code from the owner's manual of Commodore 64 to the Commodore console. The code made the good old machine to do a very simple thing: A balloon was flying on the screen, bouncing from the boundaries of the screen.

I was thrilled, even though I didn't understand the code and the flying balloon was nothing fancy. Back then, I was thrilled by the ability to write a small piece of code that made the computer to do tasks for me. That was the coolest thing in the world. And it still is.

Since then, I've come a long way as a programmer. Programming is still absorbing, but more and more I've started to become interested in software development in a wider sense. In this thesis, it was fascinating to study how software developers work as a team and how they collaborate and communicate with customers. I was very fortunate in being able to write my thesis about a subject that is close to my interest.

I want to thank my professor Tassu Takala for providing high-level guidance, and my instructor Risto Sarvas for encouraging and helping my out when ever I created a roadblock in my head.

I want to thank Futurice for supporting the development of the prototype. Also, thank you Futurice for arranging the Thesis camp and providing me a calm and peaceful working environment in the Master's Thesis room.

A special thanks goes to the whole Hannotaatio team that is Marja Käpyaho, Antti Vuorela, Martin Richter, Kim Dikert, Yu Shen, Maksim Luzik and Peter Lunberg. The team was incredible and without you there wouldn't be a feedback tool cool like Hannotaatio. Also, big thanks to Ville Saarinen, who was the Hannotaatio project customer from Futurice. Ville provided great help with the requirements gathering and gave insightful feedback.

Thank you to all the people I interviewed for this thesis.

And of course, thank you Katharina, my loved one, for all the support and encouragement.

Espoo, 8.5.2013

Mikko Koski

Contents

Abbreviations

API	Application Programming Interface
CAT	Compensatory Adaptation Theory
CSS	Cascading Style Sheets
DOM	Document Object Model
IM	Instant Messaging
MFT	Media Fitness Theory
MNT	Media Naturalness Theory
MRT	Media Richness Theory
MST	Media Synchronicity Theory
SI	Social Influence
UI	User Interface
URL	Uniform Resource Locator
UUID	Universally Unique Identifier
VCS	Video Conferencing System
XP	Extreme Programming

1 Introduction

Agile software development in its essence is all about feedback. The core principle in agile development is to have short iterations and deliver a potentially shippable product increment after each iteration (Schwaber, 2009). The product increment has to be tangible and it has to work end-to-end so that the customer is able to try it and evaluate the result of the iteration. As the customer is able to evaluate the result, she can also give feedback about the outcome of the iteration. Feedback from the customer is a powerful way for the customer to direct the product and the development team to the desired direction.

Since the rise of agile software development methods, customer communication and collaboration have been taken seriously. It has been also identified as one of the key elements in successful software projects. Intense customer collaboration over contract negotiations is one of the four Agile Manifesto cornerstones (Beck et al., 2001). In addition, in previous research it has been shown that a lack of communication and customer involvement is one of the major challenges faced by agile development teams (Korkala et al., 2006).

Agile software development principles emphasize intense customer communication and face-to-face conversation. One of the twelve Agile Manifesto principles states that "the most efficient and effective method of conveying information to and within a development team is face-to-face conversation". (Beck et al., 2001) As a result, the first version of extreme programming (XP), which is one of the agile software process frameworks demanded an on-site customer to support face-to-face communication (Beck and Andres, 2004). However, this requirement has been removed and replaced with a practice called *real customer involvement* where the customer should be involved weekly (Korkala et al., 2006).

Face-to-face communication, even though being an effective communication method, comes with a price. Face-to-face conversation requires shared time and physical location. To overcome this cost DeRosa et al. (2004) have pointed out that organizations are relying more heavily on virtual teams to save time and travel expenses. In addition, virtual teams are needed in order to integrating the work of specialized employees who might be geographically dispersed.

As stated above, there seem to be two conflicting requirements. Organizations are moving towards virtual teams even though the new agile methodologies demand intense customer collaboration, preferably face-to-face. The situation creates a need to research new ways how these elements can be combined. The need has been identified in a previous research. Korkala et al. (2006) underline that communication and feedback mechanisms should receive special attention in agile development when there is no on-site customer.

A lot of research have been conducted about communication in software projects but only a few with a focus on some specific aspect of communication, for example feedback communication. I believe that customer communication in software projects is a wide subject that includes different types of communication methods in different situations. For example, the communication media and methods required while doing planning, design or requirements gathering is very different from

the communication practices required when a customer is giving feedback. Thus, there is a need to research what the communication media are that fit best for customer feedback in agile software development.

1.1 Research objectives

The communication is an essential part of agile software development. Moreover, the feedback given by the customer to the development team is in a crucial role in order to make the project successful. However, the tools the customers are using to give the feedback have not developed much further since the development of today's e-communication tools such as Skype or video conferencing systems. My strong assumption is that most of the feedback is still given via traditional communication tools such as email, phone or face-to-face.

With this thesis, I want to experiment and validate a new kind of feedback tool prototype, which could boost the amount and quality of the feedback provided from software project customers to the developers. The objective is to generate understanding of a properties that are essential for feedback communication tools. This knowledge can be utilized in the future for new feedback tool development.

1.2 Research question

With this thesis, I want to answer to the following research question:

- *What are the properties that make a communication tool effective for giving and receiving feedback in agile software projects?*

1.3 Scope

The focus in this thesis is on feedback communication. Various researches exist about communication in general in agile software projects. I believe that customer communication in software projects is a wide subject that requires focused research on each subsection. Different communication situations require different interaction between the customer and the development team.

This thesis concentrates on software projects and especially on agile software projects. The reason for this is that agile software projects require a lot more collaboration and feedback compared to waterfall style projects where the emphasis is on contract negotiations (Larman, 2004). In agile software development, the project goal is only vaguely defined in the beginning of the project. With extensive communication and collaboration between customer and the developer, the goal of the project is crystallized while the project goes on. Feedback has a special role in this collaboration. In iterative development, the team delivers a small increment to the product frequently. The customer is then able to try the newly deployed increment, give feedback and with the given feedback help the team to reach the common project goal. Because of the special role of feedback in agile software development, it makes sense to scope the thesis only to agile software projects.

As mentioned, the main idea of agile and iterative development is to release small increments frequently thus allowing rapid feedback loop. However, feedback can be given about number of thing in software projects. For example, the customer can give feedback about concept design, user-interface graphics, project practices, project process or the delivered working piece of software. Because the deploy–feedback loop is in a key role in agile development, the feedback about the working piece of software is in the focus in this thesis. Also feedback about the side products of the software, such as concept design and user-interface specification are discussed, but the software product, namely the deployed product increment, is in a key role in this thesis.

The project team can acquire knowledge and feedback about their product from various sources. The end-users are a source of extremely valuable feedback. In fact, the feedback from the real end-users is probably the most important feedback the team can get. However, in this thesis I am not focusing on end-user feedback. Instead, the focus is on feedback from the project customer to the development team. The customer–team feedback communication can be established before the product is release for production, but the feedback communication between the end-users and the team requires the product to be available for the end-users. In this thesis, I do not limit the feedback communication only to prelaunch communication between customer and the team, but the focus is on that.

Communication is extremely social human-oriented activity. However, in this thesis the social perspective of communication is not in the focus. Instead, this thesis concentrates on the non-social properties of the communication media. For example, instead of asking, what are the properties in a communication tool that make the customer and developer to bond or what are the properties that make the customer to feel that she has been served with a high level of customer service, I ask, what are the properties in a communication tool that make feedback trasmitting from the customer to the developer efficient in order for the developer to act accordingly.

1.4 Methods

The methods used to answer the research question are following: First, previous work in the field of communication and media is studied. From the existing literature, a theoretical background about the problem is gathered. Second, based on the previous work, a prototype of a new feedback tool called Hannotaatio is evaluated based on the theoretical framework. Last, the prototype is evaluated using qualitative methods, namely semi-structured interviews.

The *design-science* paradigm is used in this thesis. In contrast to *behavioral-science*, which seeks to develop and verify theories, design-science seeks to extend the boundaries of human and organizational capabilities by creating new and innovative artifacts. According to Hevner et al. (2004), design-science creates and evaluates IT artifacts intended to solve identified organization problems. As stated by (Nunamaker Jr and Chen, 1990) the process of implementing the IT artifacts can provide researchers with insights into the advantages and disadvantages of the concepts. Moreover, the empirical studies and the evaluation of the proposed solution

can only be performed after the artifact has been built.

In this thesis, the organizational context is Futurice Ltd[1], a Finnish IT company. The company had identified a problem in their communication with the customers and a new IT artifact, feedback tool prototype called Hannotaatio was built with the help of a student group from Aalto University. In this thesis, the prototype is evaluated in the organization.

Qualitative methods, namely semi-structured interviews were used to evaluate the implemented feedback tool prototype. Nine people who had used the feedback tool prototype in a real-world project were interviewed.

A literature review was done in order to gather theoretical understanding about properties of effective communication media. In this paper the following communication media theories are included: media richness theory (MRT), media synchronicity theory (MST), media naturalness theory (MNT) and media fitness theory (MFT).

Studying these theories should result in understanding of the properties that are relevant for a feedback tool prototype. Understanding the properties that have an effect on the effectiveness of a communication media and understanding the reasons why people select a medium over another should help us to build the prototype solution to address the issues identified in the organizational context.

1.5 Structure of the thesis

Chapter 2 dives into the existing literature and research about communication media theories and communication in software projects. The communication media theories used in this study are introduced and the most commonly used terms are defined. The main point of this chapter is to form a theoretical background from which communication media properties can be evaluated.

Chapter 3 introduces the Hannotaatio feedback tool prototype. The design and implementation of the prototype are discussed in detail. In chapter 4 the prototype is evaluated using the media theories studied in chapter 2.

Chapter 5 discusses the research methods in depth. In addition to the literature review and theoretical evaluation of the prototype tool, this thesis uses qualitative methods to answer to the research question. Nine people who have used the prototype in a real-world software projects were interviewed in a semi-structured manner.

The interview results are presented in chapter 6. Chapter 8 concludes the results of the study, discusses about the limitations of the research and proposes further research subjects.

[1]Web, mobile and enterprise solutions, A lean IT company - Futurice, http://www.futurice.com

2 Literature

The literature studied is introduced in this chapter. The literature review focused on communication media, namely the four theories included in this thesis. The four theories are media richness theory (MRT), media synchronicity theory (MST), media naturalness theory (MNT) and media fitness theory (MFT).

In addition to communication media literature, software development literature was also reviewed to form background knowledge on the context in which the study is conducted. The focus in software development literature is on the most recent studies about modern agile software development methods. Older literature descriping non-agile methods is also briefly covered to provide some historical background.

The main sources of literature have been electronic scientific article collections. In addition, the Futurice company library provided a good collection of books about agile software development.

2.1 Definitions

This section defines the terms commonly used in this thesis. This is especially important, because in the thesis I am using term that do not have clear and commonly agreed definitions.

For example, *Agile software development* has been a hyped buzzword in the field of software industry. However, there is no clear definition for agile software development. The Agile Manifesto (Beck et al., 2001) only lists principles for agile development, but it does not provide a clear definition of what agile development is. Due of the lack of the definition, it is often perceived that the term is widely misused for marketing purposes by companies that are not actually doing agile development (Singleton, 2012).

As another example, the main propose of communication can be argued to be information and knowledge sharing. However, communication can lead to other positive effects, such as building a bond between communication participants. Because the positive effects of communication are manifold, it is necessary to define what communication effectiveness means in this thesis.

2.1.1 Customer feedback

In this thesis *customer* refers to a person in a software project who is the feedback provider. In the context of an external software project, customer refers to a representative from the customer organization who gives feedback to the supplier organization. In the context of an internal software project, customer refers to a representative in the internal organization who is in response of the project outcome and who provides feedback to the development team.

As the context of this thesis is agile software development, by customer I refer to a Product Owner. A Product Owner is an agile team member who is responsible for the outcome of the product. The Product Owner maintains the product backlog and scope and prioritizes the items in the project backlog. Because the Product

Owner is the one who is responsible for the project outcome, she is also a person who most likely provides the team with the most valuable feedback (Pichler, 2010).

Feedback is a part of the communication between the customer and the software supplier. It is a phase of customer–supplier communication that can happen only after the supplier has delivered something concrete to the customer. Obviously, there is very little for customer to give feedback about, if the supplier has not delivered anything yet. Thus, it can be argued, that feedback is a form of communication that happens only after the project has been going on for some time.

Feedback can be given about various subjects in software projects. Feedback can be given for example about the working practices, working processes, design documents, user interface drafts or working piece of software. In this thesis, the focus is on the feedback given about the working piece of software that has been delivered to the customer. There are various ways how software can be delivered from the development team to the customer (for example DVDs, email etc.). However, in agile software projects the preferred way to deliver software to the customer to test is to allow the customer to access to a Continuous Integration or a staging server and enable a build system that automatically creates nightly builds (Shore and Warden, 2007) (Beck and Andres, 2004). In this thesis it is assumed that the feedback is given about software that is running or otherwise available on a testing server that is updated real-time as the development goes on.

2.1.2 Effective communication

Communication is in key role in today's agile software development. Intense communication between the customer and the development team leads to various benefits. The most important and the most obvious benefit of communication is information and knowledge sharing. However, other benefits, such as social impact of communication, cannot be understated. Through communication the customer and the development team can build trust and team spirit. These aspects are highly valuable in order to increase the motivation of the team and the satisfaction of the customer.

Because the benefits of communication are manifold, the meaning of effectiveness of the communication media can be ambiguous. While one communication media can be effective in sheer information or knowledge transfer other media can be powerful in building team spirit and bond between the customer and the developers.

Even though I recognize the importance of the social aspect in communication, the main scope in this thesis is on the more result-oriented properties of communication performance. In this thesis the effectiveness of communication media refers to medium's ability to transfer information from an individual to another in order to achieve a mutual understanding. From the feedback point of view this means the ability for the customer to send the feedback message to the developers so that they are able to act accordingly.

This approach to the communication effectiveness was chosen because of two main reasons. First, the social aspect of communication was scoped out in order to make the thesis focused and keep the research scoped. Second, the selected theories, MRT, MST, MST and MFT support well this approach. However, as mentioned, I

recognize the need to investigate also the social aspect in feedback communication and thus it would be valuable subject for further research.

2.1.3 Agile software development

The focus of this thesis is on software development and more precisely on *agile* software development. The term agile was officially introduced in 2001 when the Agile Manifesto was published (Beck et al., 2001). Since then, agile software development has been gaining great amount of attention and popularity in the field of software industry. The traditional waterfall processes have been replaced with agile methodologies in many organizations and the most recent studies show that the change is not only a passing fad (Laanti et al., 2011). Instead, it has been shown that agile methodologies have beneficial effects, such as higher satisfaction, a feeling of effectiveness, increased quality and transparency, increased autonomy and happiness, and earlier detection of defects (Korhonen, 2012).

Traditionally, the so-called *waterfall*-style process model has been predominant in the field of software industry. The waterfall model was first introduced by Royce (1970). The name waterfall comes from the sequential nature of development work. Each work phase follows each other as illustrated in figure 1. The most common work phases are requirements gathering, design, implementation, testing and maintenance.

Waterfall model has been extremely challenging from a feedback perspective. Due to the sequential nature of the process the testing and the operation phases happen after the implementation phase is completed. Depending on the size and complexity of the software, the implementation can take from several months to several years. In practice, this means that the development team may have to work months after months without receiving feedback about the product. In addition, because the product is taken into operation only after the testing phase is done, the development team might never receive any feedback from the real end-users of the product before the maintenance phase.

Waterfall development relies heavily on contract negotiations. Contracts with fixed scope, time and budget do not spur feedback communication between customer and supplier. The reason for this is that there is no need for communication or feedback during the development, since all the details have been agreed and written down to the contract. I argue that the main purpose of some waterfall projects, which relies on heavy contracts, is not to build great products, instead, to build products that fulfill the contract.

Before a fixed contract project starts, the requirements for the product are gathered together with the customer and the supplier. After that, a contract is signed between both parties. The supplier promises to deliver working piece of software with the agreed features due the given deadline. In general, no changes can be made to the project scope after the signing of the contract. However, if the customer wants to add new features to the contract or remove existing features from the contract, a high cost "change request" has to be made (Beck and Andres, 2004).

From a feedback point-of-view, in the worst case the customer may not have

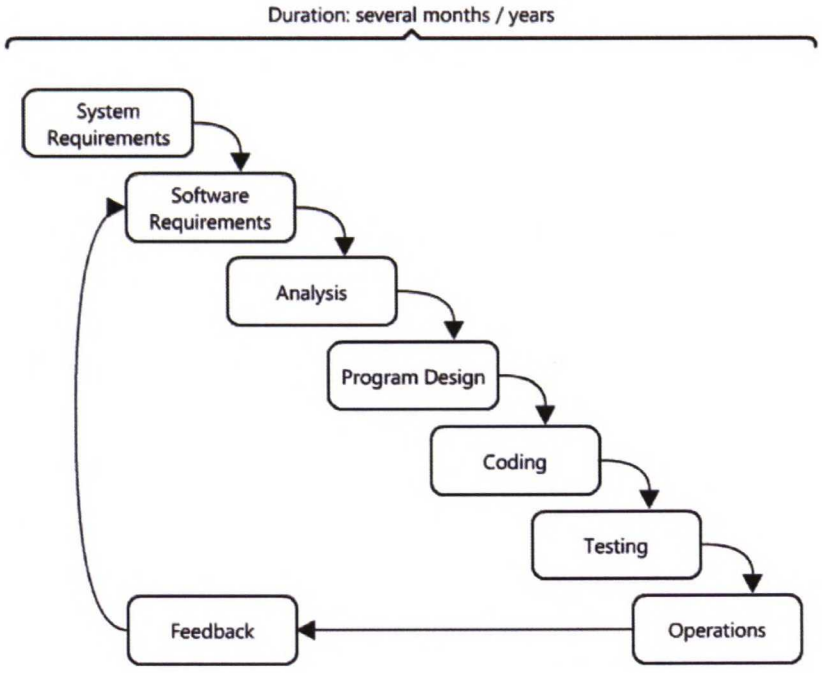

Figure 1: Waterfall process model and feedback. Reconstructed and modified from Royce (1970)

any visibility on the actual product before the agreed delivery deadline. Thus, the customer is unable to provide any feedback during the development and testing phases. Due to the lack of feedback during the implementation and testing, the customer may face unwanted surprises when the product is delivered.

To address this issue, agile development relies on intense communication and fast and frequent feedback cycles between the customer and the supplier. Instead of long-time fixed scope and fixed time contracts, agile development relies on negotiated scope contracts as a mechanism for aligning the interests of supplier and customer to encourage communication and feedback (Beck and Andres, 2004).

Instead of sequential type of process, the work in agile development is done iteratively. Every iteration contains all the phases introduced in the waterfall process from requirements gathering to testing. However, the duration of an iteration is measured in weeks, not in years. At the end of the iteration, a potentially shippable software increment is delivered to the customer (Shore and Warden, 2007).

The planning and requirements gathering in agile processes is done before each iteration. The amount of planning is minimal. The detailed planning is done in such a manner that just enough is planned to complete the next iteration. The planning in agile processes is also reactive. Based on the feedback from users, the results of user testing, the feedback from the previous iteration or the changes on the business environment, the plans can change during the project. Figure 2 visualizes

the iterative process of the development and how feedback of the delivered product affects the future versions.

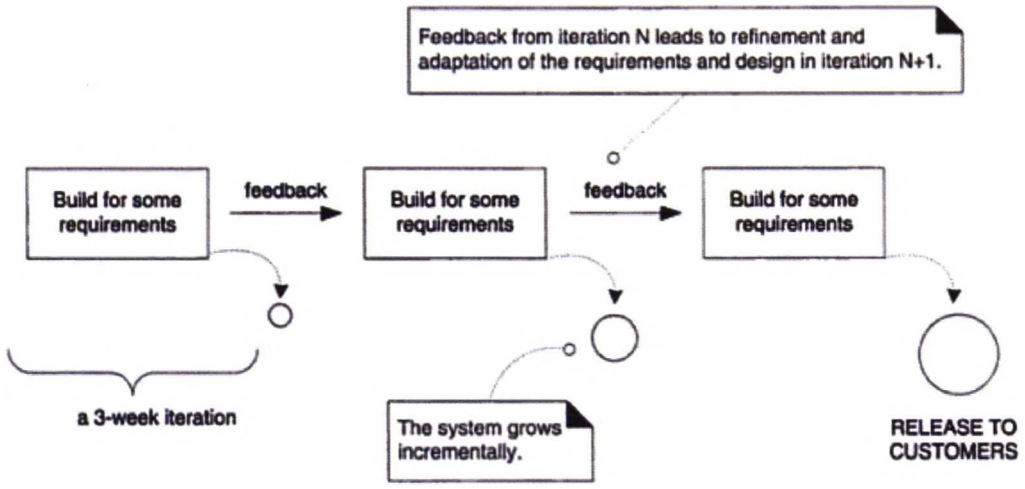

Figure 2: Iterative and incremental process relies on customer feedback between the iterations (Larman, 2004)

Iterative development opens new possibilities to the customer to give feedback. After each iteration, the customer is able to try out the deployed product increment, give feedback about it and thus guide the development team. However, this is not only an opportunity, it is also a requirement. Successful agile project demands extensive amount of feedback from the customer.

In this thesis, the key characteristic about agile development is that the development is iterative and an increment to the product is deployed after each iteration. There are various different agile process models, such as *Scrum*, *extreme programming* and *Kanban*, but from the perspective of feedback communication and this thesis, the details of the process model do not make much difference, as long as the model is iterative.

2.1.4 Agile development team

In this thesis, it is often said that feedback is given from customer to the *development team*. In the context of this thesis, the development team refers to an *agile development team*.

An agile software development team differs significantly from a traditional development team. First, an agile team is smaller than a traditional development team. There has been a lot of debate about the optimal team size for an agile development team. In general, the optimal size is commonly said to be less than ten people. Second, an agile team is interdisciplinary. Instead of having a separate requirements engineering team, a user interface (UI) design team, a development team and a

testing team, an agile team consist of a mix of people with different skills. Third, an agile team does not usually have a leader or manager. Instead, the team is self-organizing (Cockburn and Highsmith, 2001). However, as Cockburn and Highsmith (2001) states, this does not mean that they are leaderless. Instead, they are a team that can organize again and again, in various configurations, to meet challenges as they arise.

The fact that the team does not have a clear leader role is interesting from the feedback and communication point-of-view. Instead of having a communication proxy, namely the project manager, between the customer and the developers, the aim in agile development is to remove communication bottlenecks like this. Removing the bottleneck reduces the cost of moving information between people (Cockburn and Highsmith, 2001).

In agile development, the Product Owner, which is usually the project customer, is in charge of keeping the project backlog in priority order. This means that when the team receives feedback from the product end-user, customer's customer or from the customer herself, the Product Owner has the power and responsibility to choose *which* ones of the received feedback comments need to be taken into account and *when*. However, the agile team has high level of autonomy. The team is responsible for deciding *how* to answer to the feedback.

2.2 Media theories

A great amount of research has been conducted about communication media and a number of theories have been formed. In order to form theoretical background about communication media, four theories have been selected to this thesis. The selected theories, media richness theory, media synchronicity theory, media naturalness theory and media fitness theory, are explained and discussed in details in this chapter.

Including these theories to the thesis gives in depth knowledge about the communication in agile software development. Examining only the agile methodologies would not result in the same amount of knowledge, since communication is only one part of agile methodologies. Even though agile methodologies emphasize intense communication, they do not give much guidance on how the communication should be organized effectively. I believe that focusing on the feedback communication with the help of the communication theories can give new insight that could not be possible to acquire only by examining the agile methodologies.

However, examining the agile methodologies with the media communication theories is not completely unique idea. Previous research has been conducted about communication in agile development projects using the media theories. For example, Korkala et al. (2006) utilized MRT to examine communication in agile environment. However, MRT has its own limitations, discussed in further sections. Due to the limitations in MRT, it makes sense to examine agile projects using more modern theories that address the shortcomings of MRT.

Even though *media richness theory* has its shortcomings, it was chosen to this thesis, because it is the most cited and the most widely known communication

media theory. The theory was formed by Daft and Lengel (1986). However, since it was first introduced in the era before electronic communication media, it has some weaknesses what it comes to new media.

Media synchronicity theory was selected because it tries to fill the gaps left by MRT. It can be also applied to explain media selection and the different properties of the communication media that make them better than another.

Media naturalness theory has an interesting viewpoint to the communication media effectiveness. It bases its main argument to Darwinian evolutionary theory stating that face-to-face communication is the most efficient and natural communication media due to our evolution. The theory was chosen to be part of this thesis, since it brings in a very different angle compared to media richness and media synchronicity theories, which are in essence very similar theories.

Media fitness theory is a rather new and interesting theory. It was developed by Higa and Gu (2007). The theory was selected to be part of this thesis because it focuses on the media selection. Media selection is especially important for this thesis. Understanding the reasons why one medium is preferred over another helps to understand what are the properties of a medium that makes it better fit for a communication task in question.

The theory of media fitness is influenced by media richness by Daft and Lengel (1986) theory and social influence perspectives by Schmitz and Fulk (1991). The theory has been empirically proven by Gu et al. (2011) to provide rather good match between the theoretical prediction of the media selection and the actual choice.

2.2.1 Excluded theories

There are also numerous relevant communication theories that were excluded from this thesis. Social influence theory and channel expansion theory are examples of two excluded theories.

Social influence (SI) theories and social aspects of communication were excluded. The main argument of SI model is that media choice and use are not only objective and rational as stated in work prior to formulation of SI model (Fulk et al., 1987). Instead, in organizations, media properties such as richness are positioned to be subjective, influenced by attitudes, statements and behavior of others in the workplace. Schmitz and Fulk (1991) admit that the relative objective features of communication media do affect how individuals perceive the media and its efficiency. However, the objective features are only one part of the communication media effectiveness.

The reason why SI model was excluded is two-folded. First, to scope the thesis, this research focuses on the objective properties of communication media, not on the social factors affecting media selection. Second, the objective is to support development of new communication media for customer feedback. The new communication media development benefits more from a research that concentrates on media properties than the social factors.

Channel expansion theory is built on top of MRT and SI model. The main argument of the theory is that communication effectiveness builds upon four knowledge-building experiences that are particularly relevant: experience with the channel, ex-

perience with the messaging topic, experience with the organizational context and experience with communication participants. The theory proposes that developing these knowledge bases will lead to communication that is more efficient. (Carlson and Zmud, 1994) (Carlson and Zmud, 1999)

Even thought the knowledge bases listed in channel expansion certainly have effect on perceived value of the communication channel, the theory takes very little stand on properties of the communication media that will lead to development of the knowledge bases. Because the focus in this thesis is on properties that make communication media efficient, the channel expansion theory was excluded.

2.2.2 Media richness theory

The theory of media richness was proposed by Daft and Lengel (1986). The theory is well known and widely supported. However, it has been facing a lot of criticism (for example (El-Shinnawy and Markus, 1997) (Dennis and Valacich, 1999) (Korkala et al., 2006)). New theories, such as MST and MNT have been formed as a result to the criticism MRT has faced.

MRT asserts that based on the capacity of communication media to facilitate shared meaning, the media can be classified either high or low in their *richness*. In order of decreasing richness, the media classifications are face-to-face, telephone, written personal documents such as letters or memos, impersonal and unaddressed written documents such as fliers or standard formal reports. The hierarchical classification is illustrated in figure 3.

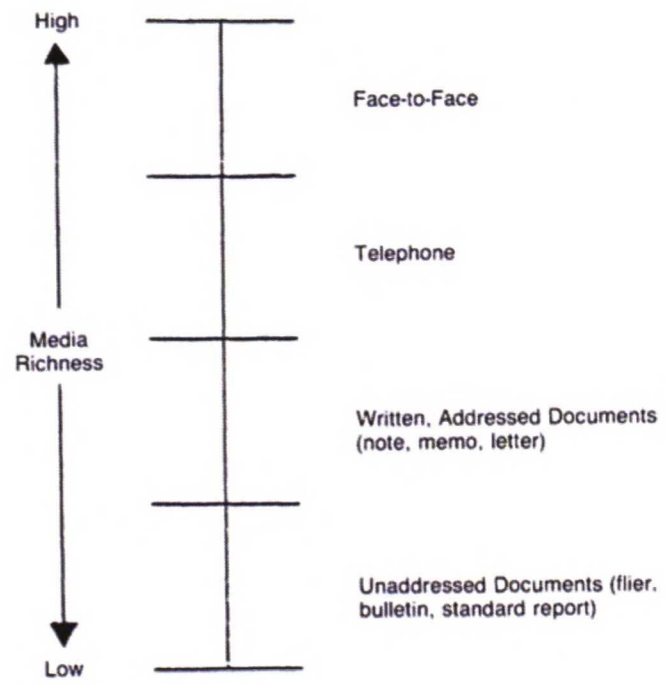

Figure 3: Hierarchy of media richness (Daft et al., 1987)

The theory also utilizes a concept of message uncertainty and equivocality. **Uncertainty** exists if information can be interpreted unambiguously but there is a lack of information. In other words, uncertainty has come to mean *absence of information.* Uncertainty has also been defined as the difference between the amount of information required to perform the task and the amount of information already possessed by the organization. Uncertainty can be reduced by acquiring more information to support the decision-making. Managers in organizations can for example simply ask questions to gain more knowledge and thus reduce the uncertainty (Daft et al., 1987).

In contrast, **equivocality** means ambiguity, the existence of multiple and conflicting interpretations, even though the amount of information available is sufficient (Daft et al., 1987). Equivocality means confusion and lack of understanding and it cannot be reduced by acquiring more information. Gathering more information may be even impossible since the managers may not be certain what questions to ask. The higher the level of equivocality is, the more negotiation is required to reach a consensus on one interpretation.

MRT lists four criteria, which define the richness of the communication media. The criteria are feedback, multiple cues, language variety and personal focus. Even though MRT does not include new online media, Graveline et al. (2000) have extended the theory and the four criteria to include the new online media. According to Graveline et al. (2000) the four criteria are described in the table 1.

The main argument about media selection according to MRT is that certain communication media are more suitable for certain task depending on the richness of the media and the level of uncertainty and equivocality of the message. A richer media is preferred for high equivocal tasks while leaner media are more suitable for tasks with low equivocality (Daft et al., 1987).

MRT is the most cited communication media theory and it continues to be the predominant theory in the field of communication research. However, it has been widely criticized. One of most remarkable shortcomings of the theory is that it was put forth before the development of the most recent electronic communication media innovations. Even today, the theory has not yet accounted for many of the "super-rich" technological media, for example virtual reality software and technology that utilizes extremely rich combination of audio, video and visual streams. (DeRosa et al., 2004)

DeRosa et al. (2004) also points out that even though MRT categorizes the media from "lean" to "rich" where the richest medium is face-to-face communication, the theory does not explain the reasons behind the superiority of face-to-face communication. According to DeRosa et al. (2004) the scale of media richness has also some flaws. As team members become more familiar and more comfortable with media lower in richness, their perceptions towards the media continued to become more positive, which increased the perceived richness of the media. In addition, the theory does not account for team member familiarity or contextual factors such as norms for technology use.

Table 1: Four criteria to define the media richness (Graveline et al., 2000) (Daft et al., 1987)

Media character	**Description**
Feedback capability	How quickly communication participants can react to the transmitted message for example by asking questions and making corrections. The capability of feedback relates to synchronicity of feedback. Face-to-face communication has high feedback capability where as exchanging documents has low feedback capability. Online media can be either synchronous or asynchronous. Synchronous media, for example videoconference have high feedback capability. Asynchronous media, such as bulletin board and email have low feedback capability.
Availability of multiple cues	The richness of various communication channels available to the participants for example physical presence, body language and voice inflection. Some online media are capable of transmitting multiple cues (for example videoconference) while some are primarily single-channel (email, text chat)
Language variety	The range of meanings that can be conveyed with language symbols. Numbers convey greater precision of meaning than does natural language. Natural language can be used to convey understanding of a broader set of concepts and ideas
Personal focus	Level of individual attention and personal feelings the message contains

2.2.3 Media synchronicity theory

Dennis and Valacich (1999) have criticized MRT for various reasons. Even though MRT has had some empirical support, various empirical researches have shown evidence against it or only partially supporting it (Dennis and Kinney, 1998) (El-Shinnawy and Markus, 1997). For example, El-Shinnawy and Markus (1997) noticed in their research that email was superior in all communication tasks even though it is ranked low in richness. They also claimed that in a situation where email was used as suggested by MRT, the reasons to use email had less to do with email's richness than with user's communication roles and medium features unrelated to the richness construct. Similar kind of results was achieved by Korkala et al. (2006) as they noticed email was commonly used communication media even though "richer" communication methods were encouraged. In addition to only partial empirical support, Dennis and Valacich (1999) strongly claim that the communication media cannot be ranked on a linear scale from "poorest" to "richest".

To fill the gaps left by MRT, Dennis and Valacich formed a theory of Media

Synchronicity. In the theory, they list five communication media properties that have an effect on communication. The properties are transmission velocity (also known as immediacy of feedback (Dennis and Valacich, 1999)), parallelism, symbol sets, rehearsability and reprocessability. In order to show the defectiveness of MRT they evaluated various communication methods from face-to-face discussions to written documents based on the five characteristics. The result of the evaluation did not support the "lean" and "rich" classification, which is the main assertion of MRT (Dennis et al., 2008). The five properties of MST are visualized in figure 4

Rehearsability stands for the message sender's ability to practice and fine-tune the message before sending it. For example, email and other asynchronous e-communication media support rehearsability well. Email sender can carefully consider the wordings in the message and she can change the wordings before sending the message. High rehearsability reduces the possibility for message receiver to misunderstand the message. However, rehearsability adds delays to the conversation. Because of the added delay, rehearsability lowers the support for synchronicity.

Transmission velocity is the speed of information transmission from sender to receiver. For example, face-to-face communication can transmit messages instantly back and forth. Email on the other hand can transmit message instantly to one direction, but getting the response might take some time. Thus, the transmission velocity in email is lower than in face-to-face. High transmission velocity supports synchronicity.

Symbol set describes the number of symbols available for the sender to encode the message for transmission. Face-to-face communication has a higher number of symbols available (vocal tone, body language, physical gestures) than for example written document, in which the main symbol is written text. Symbols can be either natural, such as vocal tone and body language, or less natural such as written text. More natural symbol sets support higher synchronicity, however, using medium with a symbol set better suited to the content of message will improve information transmission and processing.

Parallelism describes medium's ability to maintain multiple parallel communication sessions. Synchronous communication media such as face-to-face, telephone or videoconferencing have low support for parallelism. Asynchronous media such as email, or semi-asynchronous media such as instant messaging (IM) support multiple parallel communication sessions. For example, one can have multiple IM chat session open at the same time and the user can be active in multiple sessions at the same time. Parallelism lowers the support for synchronicity.

Reprocessability stands for ability to reprocess the message after it has been received. For example, email can be reprocessed and the meaning of the written message can be reinterpret. Reprocessability increases the understanding of the message content but adds delay to the conversion, thus it lowers the support for synchronicity.

MST states that the communication process is composed of two primary components: conveyance and convergence. **Conveyance** process means transforming new information. After the information sender has sent the information, the receiver processes the new information. The result of the information processing is that the

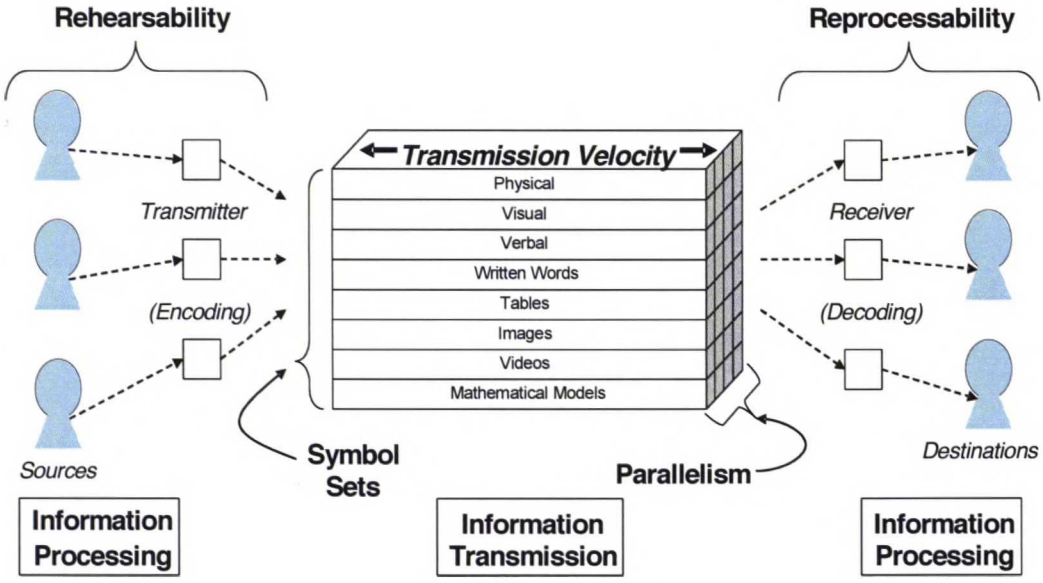

Figure 4: Communication system and media capabilities (Dennis et al., 2008)

receiver has formed a mental model of the situation. **Convergence** process means discussion and exchanging views about the information processed. The purpose of convergence is to make sure that both parties have understood the information in the same way so that their mental models are similar. The result of convergence is a shared understanding and confidence that the information was understood the same way by both parties. The individuals' familiarity with each other, the communication task in question and the communication media they are using affects the relative amount of these two processes. For familiar communication context the emphasis is on the conveyance process. (Dennis et al., 2008)

2.2.4 Media naturalness theory

As previously noted, MRT has been widely criticized and only partial empirical support to it has been found. In addition, MRT is incompatible with the SI model by Fulk et al. (Fulk et al., 1987) (Schmitz and Fulk, 1991) and thus it has been strongly attacked by social theorists (Kock, 2005). As a response, Ned Kock proposed an alternative hypothesis of media naturalness to answer to the criticism faced by MRT. Unlike MRT, the new theory is compatible with social theories (Kock, 2005).

MRT was built around the hypothesis that different communication media can be placed on a line where on one end are the "rich" media and on the other end are the "lean" media (Daft and Lengel, 1986). MNT takes a different angle to the problem and starts looking it from evolutionary perspective. The essential argument in MNT is that modern electronic communication media have evolved faster than human species and human brains. Thus, modern humans' brains are not optimally adapted for current e-communication technologies. The use of electronic communication

media increases the cognitive effort of the media use. (Kock, 2005)

There are various pieces of evidence that the evolutionary development of human species supports co-located, face-to-face communication. According to Kock, more than 99% of our evolutionary cycle humans have relied on co-located and synchronous form of communication. Facial expressions, body language and sounds, including speech, have had major role in communication. In addition, the muscles of human face have developed so that they form a complex web of muscles that allows us to use rich and expressive facial expressions. There is also evidence that the morphology of human ear suggests a specialized design to decode speech. Because of this it can be argued that humans were made to communicate face-to-face. (Kock, 2005)

MNT states that face-to-face communication is the most natural form of communication because of the evolutionary evidence. All other communication media, with worse (or better) support for natural communication elements, are less natural. The use of less natural media increases the cognitive effort. According to MNT the media can have higher support for natural elements than face-to-face communication. The use of those super-rich media also increase the cognitive effort as visualized in figure 5. (Kock, 2005) (Kock, 2004)

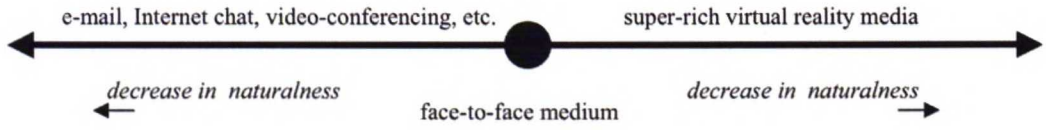

Figure 5: The media naturalness scale (Kock, 2004)

According to the MNT, electronic communication tools are less natural in comparison to the face-to-face communication. Electronic communication media have lower capability of transfering the natural elements of human communication, such as facial expressions or vocal tones.

Kock has also formed another related theory called compensatory adaptation theory (CAT). As stated by the MNT the use of electronic communication media will increase the communication effort and communication ambiguity because of the decreased naturalness of the e-communication media. The increased cognitive effort and communication ambiguity create obstacles to fluent communication. According to CAT users of the communication media will modify their communication behavior to overcome these obstacles. For example, in previous research it has been shown that telephone communication presents a significantly higher presence of verbal expressions of agreement and disagreement than face-to-face communication. The suppression of non-verbal cues, such as head nodding, was replaced by spoken words (for example, "yes", "I agree") (Kock, 2007).

2.2.5 Media fitness theory

MFT by (Higa and Gu, 2007) tries to address the mismatch between the previously

formed theories and the empirical evidence of media selection, as did the other theories, namely MST and MNT. MFT is a combination of the factors from MRT by Daft and Lengel (1986) and SI model by Fulk et al. (1987). In addition to the factors from MRT and SI, MFT adds environmental and resource limitations to the framework. By taking into account all the properties that affect on media selection, including media properties, social perspective, environmental and resource limitations, MFT tries to provide a holistic view on media selection. An overview of MFT is visualized in Figure 6.

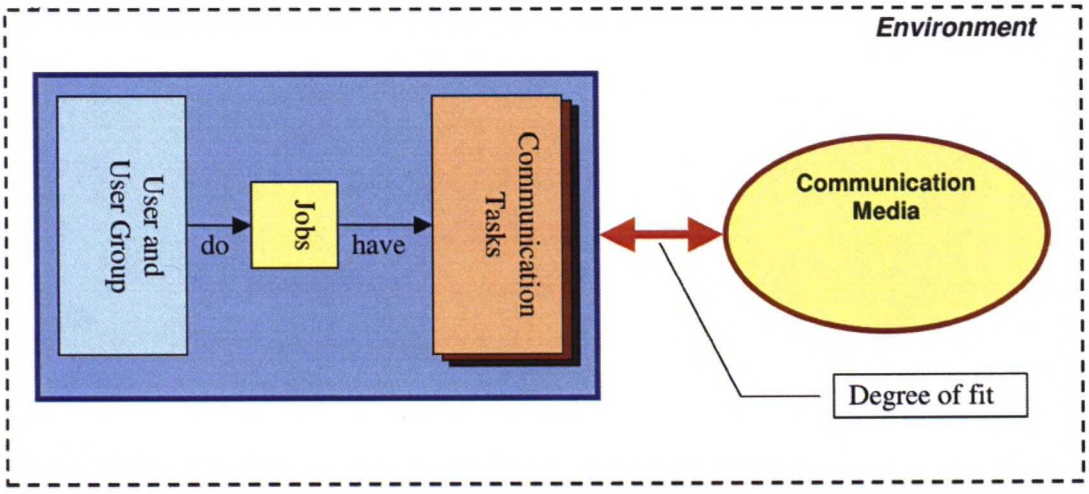

Figure 6: An overview of MFT (Higa and Gu, 2007)

The main purpose of MFT is to answer the simple question: why choose this medium but not that one (Higa and Gu, 2007). The hypothesis of the theory is that the selection is done because one medium is a better *fit* than another. The theory of media fitness defines media selection as follows: media selection is decided by the fitness of the media with the communication task needs, the communication user and user group, and the supporting environment in which the media being utilized (Higa and Gu, 2007).

MFT defines the fitness of the media by enumerating nine factors from both media and user aspect and three environmental and resource limitations. The factors are grouped into three groups.

Group I consist of communication media properties. These properties are closely related to MRT and MST. The properties in this group are response time, security, sharing, retrieval, multiparty and expressive power as listed in table 2.

Properties in group II are related to communication participants, that is, the users utilizing the communication media. The properties in this group are skill of using the media, preference of the media and the group lifespan as listed in table 3.

The last group III contains the limitations set by the environment in which the communication occurs. The properties are time availability, location availability, bandwidth and cost as described in table 4.

Table 2: Properties in group I

Name	Description
I-1 Response time	After how long an interval must the communicator get the response from the counterparty.
I-2 Security	How secure the contents of the communication should be. This issue has become more serious in computer-mediated communication.
I-3 Sharing	Whether the exact information can be shared to a third party. According to Higa and Gu (2007) the more a communication is personalized, the harder it becomes to convey the exact same message to an alternative recipient. Thus, the personalization presented by MRT by Daft and Lengel (1986) can be seen as a contradictory to sharing.
I-4 Retrieval	How easily the information may be retrieved for later use. As the amount of information transacted in organizations rapidly grows, the problem of effective indexing of information becomes serious.
I-5 Multiparty	The capability of the medium to support multiple communicators cooperating with each other by using the same medium at about the same time.
I-6 Expressive power	How many ways of encoding the message is needed by communication task. Four basic expressive powers are used: text, picture, voice and video. This property is derived from the multiple cues and language variety in MRT.

Table 3: Properties in group II

Property name	Description
II-1 Skill of using media	How well the majority of the group members master the use of media.
II-2 Preference of media	How the majority of the group members like or adapt to the use of certain media.
II-3 Group lifespan	For how long the communication group continuously exists.

The focus in this thesis is on the properties of communication media itself, that is group I properties. Group II properties describe the social environment in which the communication media is used. As noted previously, the social perspective is out of the scope of this thesis. Because of this, the group II properties are mostly

Table 4: Properties in group III

Property name	Description
III-1 Availability	The availability of medium for use. The availability is usually restricted by time and space, for example face-to-face communication can only happen in working days during office hours at the company office.
III-1-1 Available time	When the medium is available for use.
III-1-2 Available location	Where the medium is available for use.
III-2 Bandwidth	How much bandwidth can be provided for communication media.
III-3 Cost	How much money can be provided for the running of communication media by the organization.

ignored. Environmental and resource limitations described in group III are taken into account, however, less weight is put on those properties. The implementation of communication media can in some level address the properties in group III, thus they provide valuable information for the purpose of this thesis. For example, by choosing whether the implementation of the communication tool relies on video stream or asynchronous messaging between customer and developers the developer of the communication tool can affect properties such as timely availability.

The property III-2 bandwidth is ignored, since it can be argued that in the organizations of today where the companies have high speed internet connections, the bandwidth plays little or no role in media selection.

2.3 The nature of feedback communication

Before applying the selected four theories to feedback communication, we have to first define the nature of feedback communication in terms of each theory. After we have defined the properties of feedback communication, we can examine what should be the most efficient communication media, according to the selected four theories.

MRT, MST and MFT describe well what kind of communication media are the best fit for specific communication task, such as feedback. MNT provides only little description about task and communication media fitness.

In this section the nature of feedback communication is discussed and the theories are applied to feedback communication.

2.3.1 Feedback communication according to media richness theory

Before applying the MRT to feedback communication the nature of the communication task in hand has to be defined. In customer feedback communication the source

of information is the customer, who gives the feedback to the development team. In most cases, the amount of information available from the customer is sufficient for the team to execute follow up actions. In practice, this means that the customer is giving enough feedback to the team, so that the team is aware of their success and failures and the satisfaction level of the customer.

However, in some cases the customer may not give enough feedback to the team. The reason can be for example a lack of time or commitment or a lack of a person who is responsible for giving the feedback to the developers. In this case, the team has insufficient amount of information available and they have to guess how they are progressing from the customer's point-of-view. Thus, it can be stated that the level of information available in feedback conversation varies.

In MRT, the lack of information is stated to increase the level of uncertainty. When uncertainty is high, "lean" media should be used to effectively transmit the information in order to reduce the uncertainty. Because the amount of information available in feedback conversation can be either sufficient or insufficient, I argue in the sake of simplicity that the level of uncertainty in average is medium.

After receiving the feedback from the customer the development team members have to interpret the message. The feedback from customer may be very clear and unambiguous (for example "The positioning of this button is wrong"). On the other hand, the customer may be unable to provide clear and easy to interpret feedback (for example "I'm not happy with the visual appearance. I can't say exactly what's wrong with it, but I just don't like it"). In the cases like this, customer may not even know herself what is the exact message she want to transmit. Asking more questions through lean media may not solve the situation, instead a conversation is needed. Because of the partly ambiguous nature of feedback and possibility of conflicting interpretations, in the context of MRT this means that feedback communication is affected by equivocality. However, since the customer and the team are having feedback communication around a familiar and known subject (the software product) it can be argued that the level of equivocality is not the highest one, instead, medium.

The MRT proposes that "richer" communication media are more suitable for tasks with high equivocality where as "leaner" media are more suitable for tasks with low equivocality but high uncertainty (Daft and Lengel, 1986). As discussed in the previous paragraph, the uncertainty and equivocality levels of feedback communication falls somewhere in the middle. According to MRT, this means that the most effective results are achieved with communication media with medium level of richness.

2.3.2 Feedback communication according to media synchronicity theory

Majority of the feedback communication is held in a context, which is familiar to the individuals, excluding the very beginning of the project. In the beginning of the project the members are not used to the project practices nor they are familiar with the project outcome, the software product, which may not be even implemented yet. However, it can be assumed that after a short learning period in the beginning of the project the individuals have most likely gotten used to work with each other,

they are familiar with the tasks they are working on and the media they are using for communication. According to MST, in a familiar communication context the emphasis in the communication should be on conveyance process. The theory states that conveyance process is best served by media with capabilities supporting low synchronicity (Dennis and Valacich, 1999) (Dennis et al., 2008).

The theory of media synchronicity identifies five media capabilities, which define medium's support of synchronicity. Evaluating these capabilities in a context of feedback in software projects, it can be seen what kind of capabilities an effective feedback medium has. In other words, what are the properties that support low synchronicity.

Transmission velocity is the speed at which the medium is capable of transmitting the message to the recipient. From feedback point-of-view, transmission velocity is important but not as important as it is for example for novel communication tasks, such as design or planning tasks where constant and immediate interaction is required between the communication participants. In contrast of planning and design tasks, feedback is given in a context where feedback sender and receiver are familiar with the subject, that is, the software product the team is building. According to MST, when the context is familiar, the conveyance process should be emphasized. To support conveyance, communication media with lower synchronicity level should result in better communication performance. High transmission velocity supports synchronicity, thus in conclusion, for feedback purposes where conveyance process is emphasized, communication media with lower transmission velocity should be used (Dennis and Valacich, 1999).

Parallelism describes medium's ability to support multiple parallel communication sessions. Parallelism has negative impact on synchronicity. Because feedback communication requires media with low synchronicity, high parallelism, which lowers the level of synchronicity, should be preferred (Dennis and Valacich, 1999).

Symbol set describes the number of ways in which a medium allows information to be encoded for communication. More natural symbol sets support higher synchronicity, however, using a medium with a symbol set better suited to the content of message will improve the information transmission and processing. For feedback, this means that a verbal description of an activity on a web site can be less effective than a visual demonstration and a verbal description or a series of annotated screen shots with a written description (Dennis and Valacich, 1999).

Rehearsability stands for the ability to fine-tune the message before sending it. From the feedback point-of-view, rehearsability is important. An ill-advised comment from customer about an implemented feature may give a wrong impression to the developer who may end up doing a change that the customer did not actually intend. In addition, giving negative feedback to the development team in an indiscrete way may reduce developers' motivation. In conclusion, feedback communication benefits from high rehearsability.

Reprocessability is the other side of coin of rehearsability. It describes the possibility to reprocess the transmitted message. The ability to reread the message increases the understanding of the content, but adds delays to the conversation the same way as does rehearsability. The understanding of the feedback given by the

customer increases if the developer can reprocess the feedback. This is especially true if the message is communicated via a medium that does not support symbol set suited to the content of the message, such as written email when a screenshot would be more natural.

In many occasions, the received feedback requires actions. The required action may not be executed immediately. If for example a developer makes a change based on the customer feedback after a couple of days after receiving the feedback, the reprocessability plays a great role. For example, if the customer gives feedback in a face-to-face meetings (for example "change the color of the button to bluish green"), the developer may have forgotten the details of the feedback when she starts conducting the corrective actions ("change the button color green").

In conclusion, feedback is given in a context, which is familiar to the individuals working with each other thus moving the emphasis from the convergence process to the conveyance process. According to MST, the conveyance processes are best served by media with capabilities supporting low synchronicity. Media with low synchronicity are for example written documents, fax, voice mail, asynchronous electronic mail (email) and asynchronous electronic conferencing (Dennis and Valacich, 1999). According to the results the capabilities of the most suitable feedback communication media are low transmission velocity, high parallelism, high rehearsability and high reprocessability. These results are listed in table 5.

Table 5: Media capabilities and their importance for feedback

Media capability	**Description**	**Importance for feedback**
Transmission velocity	The speed at which the information is transported from an individual to another	Low
Parallelism	Capability for multiple parallel communication sessions	High
Symbol set	Diversity of symbols, which allow information encoding. Natural symbols are vocal tones and physical gestures etc.	Low
Rehearsability	The ability to fine tune the message before sending it	High
Reprocessability	The possibility to reprocess the transmitted message	High

2.3.3 Feedback communication according to media naturalness theory

Of all the four theories used in this thesis, MNT provides the least guidelines to communication media choice per different communication task. The main argument of the theory is that communication media with the best support to properties in

face-to-face communication are the most natural, thus the most efficient communication media. According to this statement, the emphasis on a communication tool which is used for feedback communication should be to increase the naturalness by supporting the natural communication properties, such as body language. (Kock, 2005) (Kock, 2004)

Kock (2007) points out that the focus in the most recent research has been on information visualization. Information visualization studies place the emphasis on extracting visual patterns from textual or numeric data. That is, the emphasis is on the development of text-to-visual representation. However, MNT proposes that visual representation is seen as more natural and thus likely to be easier generate than written text. In addition, Kock has stated that the burden of electronic media obstacles is on yhe sender's side. Thus, the emphasis on communication media development should be on the sender's side. In other words, the new electronic communication media should make it as easy as possible for the message sender to encode the message. Since visual representation is likely to be easier to generate, Kock proposes that enabling visual representation-to-text conversion could in turn significantly facilitate compensatory adaptation, thus reducing the cognitive effort of the sender.

Kock also proposes that to use electronic media effectively, managers should use combination of media in their communication interactions. As an example Kock suggests using email with video or audio clip attachment. More natural encoding mechanisms, such as video or audio, can be used to compose messages that contain large number of complex ideas. Text can be used to convey small number of simple ideas. Kock predicts that this would lead in significant reduction in the amount of text exchanged through email messages in organizations thus increasing the overall efficiency of communication Kock (2007).

2.3.4 Feedback communication according to media fitness theory

MFT provides a framework to calculate the task–media fitness. The fitness is calculated by first assessing the capabilities of the communication media according to the 13 properties defined in the theory. After that, the needs of the communication task in question are defined. The fitness can be calculated, when both the capabilities of the communication media and the needs of communication task are defined. If the media capabilities and task needs match, the communication media fits well to the communication task.

The communication task I am interested in this thesis is following: "give feedback of a software product under development". Nakamura et al. (1995) have proposed four communication types. The types are notification/transmission, coordination, creation or decision. Depending on a situation, the type of the feedback communication can be argued to be notification/transmission or coordination. For example, if the customer uses feedback to share information or knowledge to allow the development team to make corrections to the product, then the communication type is more notification/transmission than coordination. However, if the customer sees a bug in the product or a mismatch between implemented and desired user interface,

she might want to control the situation by sending a feedback to the team requiring a corrective action. In this case, the type of communication is coordination. The table 6 describes the communication task in question.

Table 6: Description of the feedback communication task

Task	**Task type**
Feedback	Notification/transmission or coordination
Task description: Give feedback of a software project to the development team	

By using the framework provided by MFT (Higa and Gu, 2007), the needs for the task can be defined. Each property is given a value from one to five describing the need for the given communication task. The MFT framework provides description for each value on one to five scale. For example, for response time the value 1 is given, if the response in two or more days is suitable. Value 2 means that the response should take maximum of one day, where as 3 stands for response in one hour or a few hours. 4 means response in 10 to 30 minutes and 5 for real-time or near-real-time response. Each property is evaluated against the descriptions provided by the framework.

The results of evaluation of the communication task needs are described in table 7.

MFT provides the similar assessment for communication media properties. The sample media definition contains six typical media for business communication: fax, telephone, email, IM, video conferencing system (VCS) and face-to-face. The sample media definition contains four values for each property: min, best-, best+ and max. The min and max mean the limits to which a medium supports the property. The best- and best+ mean the limits to which a medium can exert good support for the corresponding property. The sample media definition is shown in table 8.

To understand the table 8, the definition of properties in table 2 must be referenced. As an explanation for the sample media definitions, Higa and Gu (2007) explain the values given for email as follows: Email is a medium which is supposed to be able to respond from "10 to 30 minutes" (max, value=4) to "two or more days" (min, value=1). But email is commonly recognized as a medium which is suitable for tasks whose response time needs ranges from "one hour or few hours" (best+, value=3) to "two or more days" (best-, value=1).

Given that the communication task needs for feedback communication and the sample communication media are now defined, it is possible to calculate the fitness of the sample media for feedback communication. The media fitness framework uses the idea of fuzzy math. The fuzzy math idea simplifies the score range to 0, 0.5 or 1, where 0 represents non-match, 0.5 partial match and 1 good match. If the communication task need is in the range of best- and best+, it is considered as a good match (1). If the task need is between min and best- or best+ and max, it is considered as a partial match (0.5). Last, if the task need is not in any of the previous ranges, it is considered as a non-match (0). The sum of scores is the average

Table 7: Needs for feedback communication task

Property	**Need**
I-1 Response time	1 (response in two of more days)
I-2 Security	3 (avoid to be known by anyone except certain people)
I-3 Sharing	4 (sharable without information loss)
I-4 Retrieval	4 (semi-automatic indexing)
I-5 Multiparty	2 (about three to six people)
I-6 Expressive power	(1) Text: ABcCdD (printed, digitalized, not formatted or formatted only for easy reading, strictly formatted/structured according to certain standard, plain text, rich text), (2) Picture: ABcC (digitalized, colored, low quality/resolution, high quality/resolution), (3) Voice: aAbBcC (simplex, duplex, voice clip, voice stream, low quality, high quality), (4) Video: aAbBcC (simplex, duplex, video clip, video stream, low quality, high quality)

of all the values. The calculated fitness of sample media for feedback communication is shown in table 9.

The results in table 9 reveals that based on the group I properties the best match for feedback communication task is email and the second best match is face-to-face.

Please note that I ignored the properties of groups II and III since they are tightly linked to the group of people using the medium and the environment in which the medium is used. These properties surely affect to the media selection, but they have less weight in this thesis.

In section 4.4, the same framework is used to evaluate the match of the feedback tool prototype Hannotaatio to the feedback communication task.

		Fax	Tel.	Email	IM	VCS	FtF	Scheduler	FTP
Response time	Min	1	5	1	3	5	5	1	1
	Best-	1	5	1	4	5	5	2	5
	Best+	3	5	3	5	5	5	4	5
	Max	4	5	4	5	5	5	5	5
Security	Min	1	1	1	1	1	1	1	1
	Best-	1	1	1	1	1	1	1	1
	Best+	3	4	5	3	3	5	2	2
	Max	4	5	5	5	4	5	3	3
Sharing	Min	2	1	3	1	1	1	3	4
	Best-	3	1	3	4	1	1	4	4
	Best+	3	2	4	4	2	2	5	5
	Max	3	3	4	4	2	4	5	5
Retrieving	Min	1	1	1	1	1	1	1	1
	Best-	1	1	2	1	1	1	5	2
	Best+	2	1	4	2	1	3	5	3
	Max	3	1	4	3	2	3	5	4
Multiparty	Min	1	1	1	1	1	1	1	1
	Best-	1	1	1	1	1	1	2	4
	Best+	1	1	3	2	1	3	5	5
	Max	2	1	4	2	2	4	5	5
Exp. Power	Text	aAbcd	*	ABcCdD	ABcCdD	aABcCdD	aAbBcCdD	ABcCdD	aABcCdD
	Picture	Abc	*	AbBcC	AbBcC	AbBcC	aAbBcC	*	AbBcC
	Voice	*	ABC	abc	ABc	ABC	aAbBcC	*	abBcC
	Video	*	*	*	ABc	ABC	aAbBcC	*	AbBcC

Table 8: Definition of sample media: properties in group I (Higa and Gu, 2007)

Table 9: Group I properties and the match

	Fax	Tel.	Email	IM	VCS	FTF
Response time	1	0	1	0	0	0
Security	1	1	1	1	1	1
Sharing	0	0.5	1	0.5	0	0.5
Retrieving	0	0	1	0	0	0
Multiparty	0.5	0	1	1	0.5	1
Exp. power	0	0	0	0	1	1
Match	0.42	0.25	**0.83**	0.42	0.42	0.58

3 Hannotaatio - A visual website feedback tool

In this section the prototype feedback tool Hannotaatio is introduced. First, the design and implementation process of the prototype is described. Second, the application architecture and the implemented features are presented.

3.1 Hannotaatio introduction

The result of the prototype implementation was a visual website feedback tool called Hannotaatio. Hannotaatio is an open-source project publicly available for anyone for free. The source code is available at GitHub[2] version control hosting service. The software is licensed with permissive MIT license[3].

Hannotaatio is available as software as a service (SaaS) (Wikipedia, a), which means that the software and the associated data are centrally hosted in the cloud, which in this case means Futurice's servers. This allows users to start using the service without setting up their own hosting environment. However, if there is a specific need not to use the service hosting provided by Futurice, instructions how to set up own Hannotaatio instance are provided. This allows companies that are sensitive with the privacy of their feedback data to use Hannotaatio in their private servers.

After the installation of Hannotaatio, the user is able to capture the website in which Hannotaatio is installed. The user is directed to an editor view, after the site is captured. In the editor view, the user can draw the feedback on top of the captured site. For example, user can draw arrows, highlight an area or add text on top of the captured site. After the user has drawn the feedback, she publishes it. The publishing prevents the drawing from any further modifications. It does not make make the feedback publicly available, but instead the user gets a secret uniform resource locator (URL) to the feedback. With this URL, the user or anyone else who knows the secret URL can access the feedback. User can share the secret URL to any given project stakeholder via email or any other communication tool. In addition, a notification email about the new feedback is sent to the selected people, for example to the development team.

Good usability was one of the main design requirement for Hannotaatio. This has been addressed by for example selecting technologies that do not require user to install any browser plugin. In other words, the customer does not have to install anything on her computer in order to be able to use Hannotaatio. This lowers the barrier to start using Hannotaatio for the first time.

A special kind of site capturing technique was used to accomplish the capturing without any browser plugins. Taking a screenshot image requires external browser plugins, such as Java applets or browser add-ons. However, Hannotaatio does not take screenshot, even though the result looks like a screenshot image. Instead, Hannotaatio saves the current state of the document object model (DOM) tree of the page and the associated cascading style sheets (CSS) styles. The DOM source

[2]https://github.com/futurice/hannotaatio

[3]http://opensource.org/licenses/mit-license.html

and the CSS styles are send to Hannotaatio server where they are stored. When the user views the feedback, the site is rerendered with the saved source.

3.2 Design and implementation of the prototype

The prototype was built by a student group from Aalto University for a Software Development Project course. I was part of the student group and my role was a project manager and an architect. Besides me, the team included seven other students, who are studying either Computer Science or Information Networks.

The course was three periods long, that is, from September to February. The course was splitted in three sprints: planning, implementation I and implementation II. During the planning sprint the requirements for the software were gathered and agreed with the project customer. In addition, the initial software architecture design and the user interface design was completed. The software implementation was made during the two implementation sprints.

A Finnish IT company Futurice was the sponsor of the student project. Futurice is a lean IT development and consultancy company, which provides web, mobile and enterprise solutions and digital strategy consulting. The project customer was a Futurice employee. The customer helped the team to prioritize features and was the Product Owner of the project.

3.2.1 Problem setting

Before the project started, the company had identified a problem in their feedback communication with the customers. The problem was that the customers were using inefficient tools for giving feedback. They were for example writing long emails describing textually parts of the application that would have been easier to point out visually. The textual description of an application feature is both heavy for the feedback sender and hard to interpret for the feedback receiver. It also increased the possibility of misunderstandings.

On the other hand, some clients were actually using visual feedback by first taking screenshot of the application, then attaching it as a part of a Microsoft Word[4] or Microsoft PowerPoint[5] document and then drawing arrows on the screenshot in Word or PowerPoint. After the drawing was ready, the document was sent to developers as an email attachment.

Even though the visual aspect of this method added a lot of value, there were clear issues. First, it was laborious for the feedback sender. Due to this, some feedback may left unsent, because the customer did not have time or interest to do the laborious work. Second, the developers were not too delighted to receive Word or PowerPoint attachments, since they are somewhat difficult to open. Some

[4]Document and Word Processing Software | Microsoft Word - Office.com, http://office.microsoft.com/en-us/word/

[5]PowerPoint Presentation and Slide Software - Office.com, http://office.microsoft.com/en-us/powerpoint

developers did not have Word or PowerPoint installed. Third, it is well known that sending Word or PowerPoint attachments is a security risk (Tyson, 2011).

3.2.2 Designing the solution

The goal of the project was to build a working production ready prototype for Futurice to try out a new method to accelerate the feedback communication in their projects by solving the problems mentioned above. The goal was to both increase the amount of feedback from Futurice's customers and make the feedback communication more specific.

During the planning sprint, the team designed the solution for the existing problem together with the customer. Before the planning started, the customer had only vague idea about the possible solution. Hanno Nevanlinna, a Futurice employee, was the father of the idea. Hanno's high-level idea was to build a tool, which "improves customer satisfaction by giving them a change to participate and communication with the project team". He had also an idea that giving feedback by drawing on top of the website could be the way to give the customer a change to participate.

Designing of the software included benchmarking other existing solutions. Markup[6] is an application for drawing on top of a website. It is very similar to what Futurice needed. It can be installed as a bookmarklet by dragging the bookmarklet installation link to the bookmark bar. After the bookmarklet is installed, user can click the bookmark and start drawing the shapes on top of the website.

However, Markup had three shortcomings which were unacceptable for Futurice. First, the customer who would use Markup would have to install the Markup bookmarklet to her browser before she is able to give feedback. As the goal of the project was to accelerate the initialization of the feedback communication, the requirement for bookmarklet installation before giving the feedback was unacceptable. The customer may leave the feedback unsent, because she does not have time or interest to install a new tool. Second, the feedback messages sent with Markup are stored in Markup's servers. Thus, Futurice or its customers would not have any control over the feedback data, stored in servers controlled by third party. For Futurice, it was a requirement that the feedback has to be stored to customers own computer or servers controlled by Futurice. Third, Markup does not capture images. Image capturing is explained in detail in Section 3.5.6, but in practice the lack of image capturing would have made the use of Markup difficult, if the server is not publicly available, that is, it is behind a firewall or secured with a password. The websites Futurice develops are run in private testing servers often protected by passwords. That is why it is important that the feedback tool is able to function also in private websites.

Skitch[7] was another benchmarked alternative. Skitch allows user to take a screenshot and annotate it. After adding the annotations, the user can send the annotated image file to the developers via email. The problem with Skitch was that as with Markup, it requires user to install the application before giving the feedback. The

[6] http://markup.io/

[7] http://evernote.com/skitch/

need to install new tools was something Futurice wanted to avoid. On the other hand, the solution Skitch is providing requires too much effort from the feedback sender, who has to first open the Skitch application, take the screenshot and manually send it to developers. This solution does not differ much from the existing situation, where customers are sending the annotateted Word documents.

Due to these limitations, Futurice was neither keen on using Markup nor Skitch to solve the problem. It was identified that these is no existing solution which would solve Futurice's problem and respect the constraints they had. However, Hannotaatio was highly inspired by the existing solutions.

The development team produced two design documents during the planning sprint: a one page document describing what the software is, what it is not and why is the team building it and another document which contains a prioritized list of requirements.

The purpose of the one page document, which was called "the A4", was to clarify the idea of the software on the high-level. The A4 was agreed together with the customer and it was used as a design cornerstone for later decisions. When ever the team or customer was unsure whether a new feature idea should be included in the software or not, the A4 was referred to. The A4 can be found from appendix B.

The purpose of the requirements list was to list all the desired features in detail in a form of a user story. A user story is a one line description which captures the desired functionality from the user point-of-view (Wikipedia, b). User stories are written in semi-structured format. In our case, we wrote the user stories in the follwing format: "User can do <action>", for example, "User can add a text label on top of the website".

The requirements list contains three types of requirements: functional requirements (FR), quality requirements (QR) and constraints (C). Functional requirements describe *what* the application should be able to do. Quality requirements describe *how well* the application should be able to conduct the desired action. For example, quality requirements can describe the desired performance of the application. Constraints describe the boundaries in which the implementation has to be made. For example, constraint can describe the environment in which the application is used. An example of a constraint could be "the application has to work in touch-screen mobile devices".

The list of requirements can be found from appendix C.

3.2.3 Project success criteria

The requirements list formed the success criteria for the project. The list was prioritized and thus the customer expected that at least the user stories with the highest priority will be accomplished. From the requirements list it can be seen that the development team was able to accomplish most of the required features. The user stories which have been marked as "accepted" were finished and accepted by the customer. The user stories which have been marked as "approved" were approved requirements by the customer but were left unimplemented by the team due to the time constraints.

Two high priority quality requirements were not accepted as it can be seen from the requirements list. The two unaccepted user stories were "The DOM tree capturing captures also the images from the website", which in practice means the image capturing discussed in section 3.5.6 and "The code has to be maintainable enough so that a developer outside the team can understand the structure and fix a bug in a reasonable time". The first unaccepted feature, which is the image capturing, was unimplemented due to the lack of time. The second unaccepted feature, which about code quality, was unaccepted because the team did not test this requirement in any way. However, close eye was kept on the code quality during the whole development phase.

Also, one constraint "The system has to work with all common browsers (IE, Firefox, Safari, Chrome) excluding IE6" was left unimplemented due to the lack of time. Hannotaatio works well in Firefox, Safari and Chrome but the support for Internet Explorer is limited.

Overall, even though some requirements were not fulfilled, the customer was satisfied about the end result of the project.

A few new features have been added to Hannotaatio after the Software Development Project Course. The image capturing feature was implemented by me as a Futurice employee after the course. In addition, the user story "Futurice developers can add email addresses that will always get notification when a new Hannotaatio is published", that is, the email notification feature, was also implemented by me after the course.

3.2.4 Implementation result

After the prototype implementation was completed, it has been used in Futurice's projects. There is no clear number how many projects have adapted the prototype but presumably, the number is between 5 and 15. Since the prototype is open for public, other companies have used the prototype in their projects. Again, no clear number is available but most likely the number is about the same, from 5 to 10.

In addition to being a great feedback tool that can be used in real-life projects, the prototype also served another important function. Hannotaatio provided access to interview people who have used it in a real-world projects. This access is utilized in this thesis and the interviews of the Hannotaatio users is the main data collection method in this thesis.

3.3 Application architecture

There are three main parts in Hannotaatio application architecture: front end capturer, back end server and front end editor.

The *capturer* script is injected to the website from which the feedback will be given. The script is responsible for capturing the website and sending it to the back end server. The website author is responsible for injecting Hannotaatio capturer to the website. After the capturer is successfully injected, a small "I love feedback" button will appears to the upper right corner of the website. When the button

is clicked, the capturer captures the current DOM tree and associated CSS style definitions. The captured DOM source and styles are sent to the Hannotaatio back end server and user is redirected to the editor site.

The back end *server* stores the captured websites and annotations, and serves them to the front end editor by request. The captured sources, such as DOM source, CSS styles and images are stored to file system. The website annotation and notification emails are saved to MySQL[8] database. The back end implements RESTful application programming interface (API). The API uses JSON format to transfer data between the front end application and the back end.

The *editor* is responsible for loading the captured website and providing the annotation tools for the user. The captured website source is rendered in an iframe container, which is an HTML element that allows browser to display another webpage inside a host webpage. A drawing canvas is created on top of the iframe container. The canvas allows user to draw the feedback and save it for sharing.

Hannotaatio is implemented using mainly JavaScript and Ruby On Rails. The editor and capturer are implemented with JavaScript and HTML5. For old browser compatibility, some parts of the capturer utilize Adobe Flash for capturing images with browsers that do not support HTML5 Canvas elements. For the drawings, the editor uses popular JavaScript library Raphael.js, which is a popular library for vector graphics in JavaScript. The back end server is implemented with Ruby On Rails.

The process of capturing the website and drawing the feedback on editor is described in figure 7. The figure also describes how the three architectural parts interact with each other.

3.4 Supported platforms

Due to the selected implementation technologies, namely JavaScript and HTML5, Hannotaatio can only be used in an environment, which supports web technologies. In practice, this means web browsers. Hannotaatio can not be used in native desktop applications written in languages like C#, C++ or Java or in mobile applications written in languages such as Objective-C, Java, C# or Symbian. However, due to the increasing interest towards web technologies such as HTML5, it is possible that with a minor modifications, Hannotaatio could be extended to software areas other than websites in a near future.

At the moment Hannotaatio is most suitable for giving feedback of desktop websites. However, with small modifications Hannotaatio could support mobile and tablet websites. At the moment Hannotaatio is able to successfully take the screen capture of the website in devices such as Apple iPad and Apple iPhone, but due to the lack of touch device support, the drawing of the feedback is not possible.

There are solutions that allow native-like mobile application development using web technologies. PhoneGap[9] is one of the most prominent frameworks that packages HTML5 application to a container, which makes the application to look like a

[8] http://www.mysql.com/
[9] http://phonegap.com/

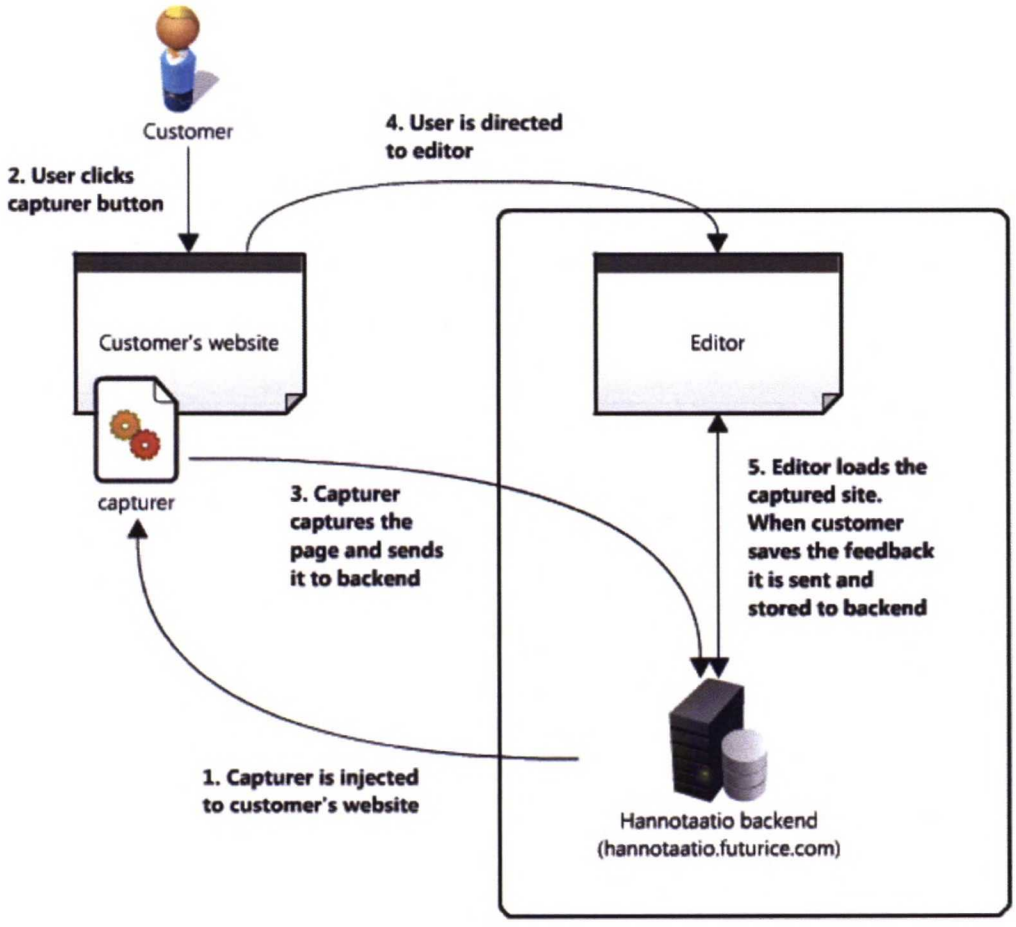

Figure 7: Capturing website and drawing process

native mobile application. The packaged application can be deployed to all major mobile platforms, such as iOS[10], Android[11] and Windows Phone[12]. Using a packager like PhoneGap, Hannotaatio could be used in native mobile applications.

Number of operating system vendors have started to support web technologies to allow developers to build native applications with these technologies. For example, Windows 8 allows developers to implement Windows Store Apps[13] with JavaScript and HTML5. In addition, the team developing GNOME[14], which is the default desktop environment of the most popular Linux distribution Ubuntu[15], has recently

[10]http://www.apple.com/ios/
[11]http://www.android.com/
[12]http://www.windowsphone.com/en-us
[13]http://msdn.microsoft.com/en-US/windows/apps/
[14]http://www.gnome.org/
[15]http://www.ubuntu.com/

unofficially announced that JavaScript will be the default language for the future GNOME application development (Reitter, 2013). As the web technologies are coming to desktop application development, it opens new possibilities to Hannotaatio. Because the technology in which Hannotaatio is implemented will be soon suitable for desktop application development, with a small modification it may be possible to extends Hannotaatio's support to desktop environments.

3.5 Implemented features

In the following sections the most important features of Hannotaatio are described. The features are: easy installation, easy initiation of the feedback process, the drawing tools, ability to share the feedback with the team, viewing the feedback and image capturing.

3.5.1 Easy installation

An easy installation process was one of the main goals for the product. Complex installation process can be a show stopping barrier for users who would like to try new communication tools but are not willing to investment too much effort for the introduction of the tool. Because of that, Hannotaatio's installation process is implemented to be as easy and fast as possible.

The installation of Hannotaatio requires minimal amount of coding and configuration. In the simplest case the website author does not have to do anything else that copy the seven line code snippet from Hannotaatio website to author's own site (Futurice, 2010).

For advanced users, there is a possibility to create an API key. The purpose of the API is to collect email addresses of the users so that they can be informed about upcoming updates and downtimes.

There is also a possibility for advanced users to change the default site capturing settings, for example, they can turn on the image capturing, which is off by default. This allows smooth use of Hannotaatio even with private websites that are protected by passwords or firewalls.

3.5.2 Initiating feedback process

Normally, before the customer can start giving the feedback, two things have to happen. First, the customer has to see and try out the product on which the feedback will be given. Second, the customer has to initiate the selected communication channel. With the traditional communication tools, such as email or telephone, the initiation process requires opening email software and creating a new email message or calling to the feedback receiver.

In Hannotaatio, a lot of work has been done to make the initiation of the feedback process as easy as possible. The solution to enable easy initiation of feedback communication channel was to add an "I love feedback" button to the upper right corner of the website from which the feedback will be given (see figure 8). This way the distance between the feedback subject and the communication tool is minimal.

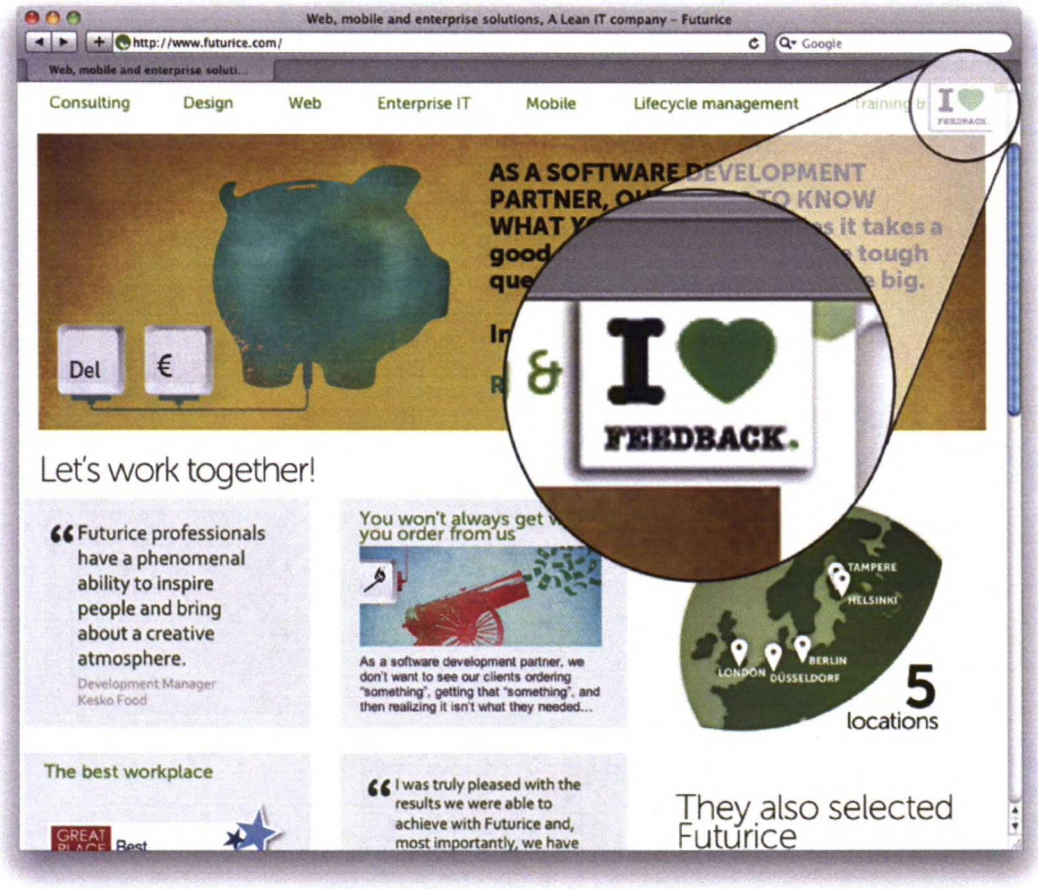

Figure 8: "I love feedback" button is added to the upper right corner of the website

When the user presses the "I love feedback" button, a screen capture is taken from the website. After that, user is directed to an editor, where user is able to draw on top of the captured website.

3.5.3 Drawing tools

In Hannotaatio, there are couple of tools for the user to draw the feedback on top of the website. The number of tools have been kept minimum on purpose to make the application extremely simple to use.

The available drawing tools are pointing arrow, rectangle and text box. In addition, the color of the drawing can be changes between dark and light color scheme. This allows user to draw on top of either light or dark websites.

In the requirements gathering phase it was identified that the three most important functions the drawing tools must allow are pointing, highlighting an area and leaving textual note. The implemented drawing tools, arrow, highlight box and text box allow all these required functions. However, other drawing tools such as free-

hand drawing tool or circle drawing tool was left unimplemented, because they were not critical tools to accomplish the desired functions of pointing, highlighting and leaving a note. The figure 9 is a screen capture of Hannotaatio's toolbar containing all the implemented drawing tools.

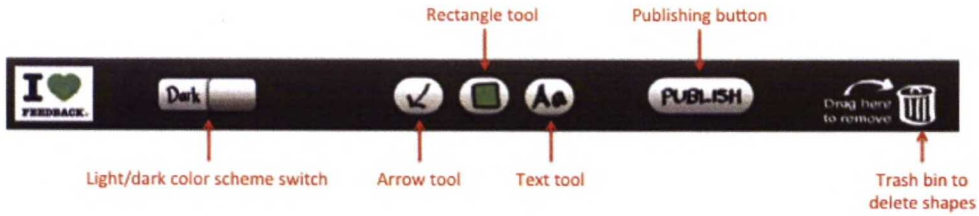

Figure 9: Hannotaatio toolbar

3.5.4 Sharing the feedback with the team

When the customer has finished drawing, the first step to share the feedback with the team is to publish the feedback by pressing Publish button (see figure 9). When the drawn feedback is published, no further modification can be made.

After publishing, user is given a secure URL, which she can share with the team for example via email. The secure URL is randomly generated universally unique identifier (UUID) and it is long enough so that it is impossible to guess. That makes it secure even though viewing the feedback does not require password or any other user credential.

Optionally, if the team has set predefined notification email addresses, a notification email is sent to the team. This happens right after the feedback has been published. If the email notifications are used, customer does not have to manually share the secure URL with the team.

3.5.5 Viewing the feedback

After the drawn feedback has been published by the customer, the development team receives a notification email with the secure URL to the newly drawn feedback or the team receives the secure URL from the customer via email.

The team can now access to the published feedback. Besides seeing the drawn feedback the team can also see when the feedback was given and which browser and operating system was used. Additionally, the team is able to access the original site from which the feedback was given by clicking "Go to original page". Also, the team or whoever has access to the secure URL can delete the feedback by pressing "Delete" button. The figure 10 is a screen capture from a published feedback from the viewer point-of-view.

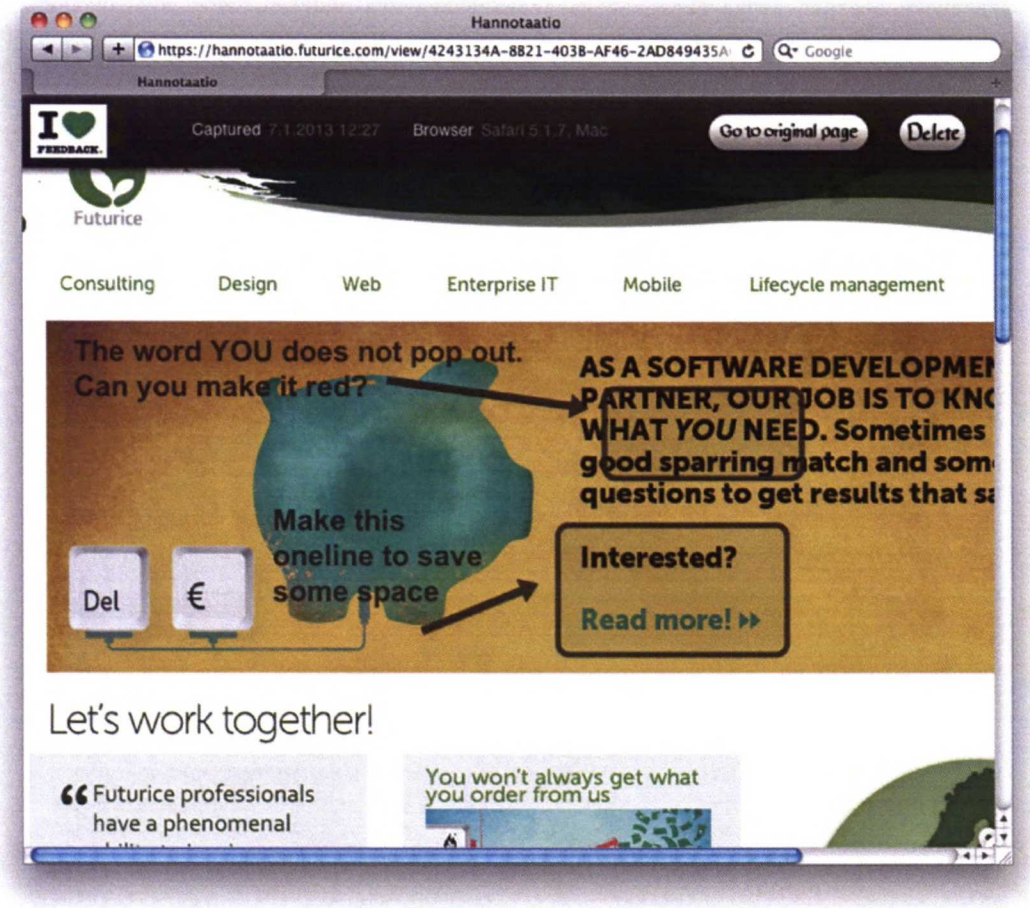

Figure 10: Published feedback

3.5.6 Image capturing

Image capturing is a key feature in Hannotaatio. The lack of image capturing in products that existed before Hannotaatio was one of the key reasons why Hannotaatio was implemented in the first place. In practice, image capturing means that in addition to the captured DOM tree, the images on the website are also captured. Without image capturing, the images in the captured website are only referred by the image location URL. If the image location is not publicly available, the referred image can not be displayed. In practice, the lack of image capturing prevents proper use of Hannotaatio in a website that is not publicly available.

Image capturing was one of the most technically challenging feature in Hannotaatio. The images are captured with HTML5 Canvas. The Canvas element encodes the binary image data to text by using Base64 binary-to-text encoding (Josefsson, 2006). After the encoding the image data is sent to the back end server. The back end server decodes the image back to PNG or JPG image. If the browser does not

support Canvas element, Adobe Flash is used as a fallback solution. An invisible Flash component which is injected to the page does the Base64 encoding of the image and passes it to the browser. The browser then sends the encoded image data to the backend server the same way it would do if the Canvas element was available.

4 Theoretical evaluation of Hannotaatio

In this section the prototype is evaluated using the communication media theories. The theories and the properties, which are used by the theories to evaluate the effectiveness of the communication media were introduced and discussed in chapter 2.2.

The theories also define which properties make the communication tool most suitable to a given communication task. The communication task in this thesis is giving and receiving feedback of a software product. In chapter 2.3 the nature of feedback communication task was defined.

Using the information from previous chapters about the feedback communication task, the communication tool Hannotaatio and the media theories, the Hannotaatio tool is evaluated with each of the theories. The results of the evaluations are presented in this chapter.

4.1 Evaluation of Hannotaatio with media richness theory

The richness of communication media in media richness theory (MRT)'s richness continuum is defined by the following media capabilities: immediate feedback, number of cues and channels utilized, personalization and language variety.

The immediacy of feedback is the weak point of Hannotaatio. In fact, Hannotaatio utilizes email to transfer the notification about newly given feedback. Thus, the immediacy of feedback is not any higher than in email. Number of cues and channels utilized and language variety is well taken into account in Hannotaatio. The fact that Hannotaatio is capable of transmitting visual picture of the website, text and symbols drawn by the feedback sender increases the number of cues available. In addition, the personalization of the message is high, since the feedback sender can freely draw the feedback as she likes.

Compared to email, Hannotaatio adds an important visual element to the message. However, if compared to telephone, Hannotaatio is worse in immediacy of feedback. In the scale of richness of MRT, it can be argued that Hannotaatio positions in between of email and telephone, being a mediocre rich medium.

4.2 Evaluation of Hannotaatio with media synchronicity theory

As noted in the previous section 2.2.3, media synchronicity theory (MST) identifies the following properties of a communication media: transmission velocity, parallelism, symbol set, rehearsability and reprocessability.

In Hannotaatio, the **transmission velocity** is low. When the development team has something to show to the customer, email is commonly used to notify the customer about the new version from which she can give feedback. After the customer has received the notification from the development team, she browses to the site, gives feedback with Hannotaatio and shares the secure uniform resource locator (URL) with the team via email.

The transmission speed of email is instant, but because getting a response to email adds some delay, email is considered to have low transmission velocity. Because there are at least two email send-receive cycles involved in one feedback, which is given with Hannotaatio, it can be argued that the transmission velocity in Hannotaatio is rather low.

In Hannotaatio, there is a possibility to use notification emails. If notification emails are used, the notification is sent to the team automatically right after the customer has published the feedback. This feature slightly improves the transmission speed because it eliminates one manual email sending from the whole feedback process.

Hannotaatio supports high **parallelism**. Because giving feedback with Hannotaatio does not require shared time and location with the feedback receiving team, customer can have many simultaneous feedback conversations at the same time. In other words, this means that the customer can give feedback with Hannotaatio at the same time when she is chatting with the team via instant messaging (IM) tool. However, it must be noted that drawing the feedback requires some concentration from the customer, so even if it is possible to have multiple conversions at the same time, it may not be very pleasant.

Hannotaatio supports also high **rehearsability**. Because the feedback is not transmitted to the developer team before the customer chooses to publish it, the customer has the ability to fine-tune the feedback drawing as long as she want. For feedback conversation, this property of Hannotaatio is important, so that customer can fine-tune the message to be as clear and understandable as possible. Because feedback can be sometimes negative, it is also good that the customer has the ability to choose the wording carefully.

Hannotaatio supports high **reprocessability**. After the customer has shared the secure URL to the development team, the team can come back to the URL which contains the message as many times as needed. From the feedback point-of-view, this property of the tool is extremely important since the team may not have time to react to the feedback immediately. For example, in agile development, it might take some weeks before the team reacts to the feedback, if the team decides to do it in the next iteration. In this case, it is important to be able to recap what was the feedback all about.

The naturalness of the **symbol set** in Hannotaatio can be argued to be medium. Visual message is more natural than for example written message. Because Hannotaatio supports visual encoding of the message (annotated screenshot) it has a more natural symbol set than for example plain text email, which (attachments excluded) supports only written message.

However, even though the message in Hannotaatio can be visually encoded, Hannotaatio misses for example vocal tones which can be transferred with for example telephone and physical gestures which can be transferred with for example video conferencing system or face-to-face. Thus it can be argued that Hannotaatio does not have the most natural symbol set, instead medium level of naturalness.

Table 10: Media capabilities and their importance for feedback

Media capability	**Description**	**Support in Hannotaatio**
Transmission velocity	The speed at which the information is transported from an individual to another	Low
Parallelism	Capability for multiple parallel communication sessions	High
Natural symbol set	Diversity of symbols which allows information encoding. Natural symbols are vocal tones and physical gestures etc.	Medium
Rehearsability	The ability to fine tune the message before sending it	High
Reprocessability	The possibility to reprocess the transmitted message	High

4.3 Evaluation of Hannotaatio with media naturalness theory

Media naturalness theory (MNT) emphasizes communication tools that are as natural as possible, that is, as close to face-to-face communication as possible. The theory lists five key elements that involve in natural communication: high degree of co-location, high degree of synchronicity, ability to convey and observe body language and ability to convey and listen to speech.

Hannotaatio, as well as the other electronic communication media, does not support these properties well. Thus, from the Media Naturalness point-of-view, Hannotaatio is not very natural communication media. However, there are elements in Hannotaatio, which make it superior in comparison to other electronic communication media, for example email.

The feedback in Hannotaatio is visual, which makes it more natural communication media than for example email or text-based chat rooms. However, Hannotaatio according to MNT can not compete with for example video conferencing system, which has significantly better ability to convey the natural cues such as speech and body language.

However, (Kock, 2007) has suggested that managers should combine the media they are using. Hannotaatio is a combination of visual representation of the feedback and email notification, thus, according to Kock's prediction, the use of Hannotaatio should lead to reduction of text exchanged though email and thus lead to an overall increase in communication efficiency in organizations.

In the future development of Hannotaatio, MNT should be taken into more careful consideration. For example, there are improvement possibilities, which would support Hannotaatio's naturalness. For example, instead of the static visual representation, the feedback could be recorded screen capture of the user's screen. In

addition, an audio narration could be added. Also, a webcam video of the user itself could be included in order to capture the facial expressions and users body language. However, in the interviews many of the interviewee said that this would not add much value to the feedback and it would increase the barrier to give the feedback. Many of them said that they would feel themselves awkward if they would have to record their face while giving feedback.

4.4 Evaluation of Hannotaatio with media fitness theory

In section 2.3.4 the needs for feedback communication task were defined by the framework provided in the theory of Media Fitness (Higa and Gu, 2007). In that section the mainly used communication tools, fax, telephone, email, IM, video conferencing system and face-to-face were used to evaluate their match to feedback communication task.

The feedback tool Hannotaatio can be evaluated with the same framework. As the needs for feedback communication task have been already defined in table 7 the same values can be used when evaluating Hannotaatio. The values of group I properties of Hannotaatio and the calculated match score are listed in table 11

Table 11: Hannotaatio media fitness theory (MFT) scores

Property	**Min**	**Best-**	**Best+**	**Max**
I-1 Response time	1	1	3	4
I-2 Security	1	1	4	5
I-3 Sharing	4	4	5	5
I-4 Retrieval	1	1	2	3
I-5 Multiparty	1	1	3	4
I-6 Expressive power	(1) Text: d (plain text), (2) Picture: ABC (digitalized, colored, high quality/resolution), (3) Voice: *, (4) Video: *			

By using the framework provided by MFT, the media fit of group I properties for feedback communication task are: Response time: 1 (match), Security: 1 (match), Sharing: 1 (match), Retrieving: 0 (non-match), Multiparty: 1 (match). The total fit, which is the average of the match points gives total match of 0.667.

When the result is compared to the results of the fitness of the traditional communication media on table 9 it can be seen that Hannotaatio places to the second fittest medium after email (0.833) but before face-to-face (0.583). The closer look to the table reveals that the properties of Hannotaatio are very close to the properties of email. This is not a surprise, since Hannotaatio closely relies on email. The link to the given feedback is usually shared by a notification email or by a manual email message from the feedback sender. From the match points between email and

Hannotaatio, it can be seen that Hannotaatio resulted with worse match points on retrieval. This is due the fact that Hannotaatio does not store the feedback URLs itself. The storing of the URL has to be done by the feedback received herself.

5 Qualitative research methods

The research methods are discussed in this section. Qualitative methods, namely semi-structured interviews, are used to answer the research question. This section provides reasoning behind the selection of qualitative methods and description of the application of the selected methods.

Based on the theoretical evaluation the implemented prototype Hannotaatio should be suitable to give and receive feedback in software projects. However, this hypothesis has to be validated with empirical evidence.

In this thesis, qualitative methods were used to validate the prototype. As Silverman (2009) stresses, the choice of methods should not be predetermined. Similarly, according to Gummesson (1999), both methods, quantitative and qualitative should be used in academic research where they are appropriate. In this thesis, quantitative methods were also considered, but qualitative method was preferred because of several reasons.

One possible quantitative method would have been to measure the amount of feedback a development team received before and after the use of Hannotaatio feedback tool. This approach would have given a statistically reliable proof whether the tool enables feedback conversation between customer and the development team. In addition, in previous research this approach has been used. For example, Rice and Love (1987) based their research about emotions in electronic communication on measuring the number of sentences, the number of messages sent and the duration of communication sessions. However, Carlson and Zmud (1999) argue that such an operational definition reflects channel use as opposed to knowledge-building experiences. In other words, more messages sent might actually mean that the message was not well understood, thus leading to worse experience on the communication channel. In addition, from the feedback point-of-view, the number of feedback given does not directly imply the value of the given feedback. By examining only the amount of feedback does not tell anything about the quality of the given feedback and the overall value of the feedback.

A structured questionnaire was another considered quantitative method. A structured survey could have overcome the problem of measuring the sheer number of feedback. With a survey, it could have been possible to ask questions about the quality of the feedback and the perceived value of the feedback. There are also a number of statistical analysis methods, which could have been used to analyze a structured survey.

However, a structured questionnaire has some drawbacks, which make it an unsuitable method for the particular case. First, surveys can give answers to questions that are known when the survey is created but they allow very poorly new questions and ideas to arise. In the particular case, I am especially interested to hear new ideas how to improve the prototype to make it even better tool for feedback. Second, questionnaires require a great number of answers to form a statically reliable sample. However, the amount of available contact information of Hannotaatio users is very limited and thus a reliable sample for quantitative methods could not have been formed.

Quantitative methods are suitable to answer questions like 'how many' or 'how much'. However, feedback communication is a complex social activity, and by examining only the 'how many' one might miss important reasons which help us to understand 'how' people behave in a communication situation (Silverman, 2009).

5.1 Data collection with semi-structured interviews

The purpose of the semi-structured interviews is three-folded. First, to better understand what are the communication media used in the organizations. Second, to ask interviewees what are the properties they value in a feedback communication tool. Last, validate whether the properties implemented in Hannotaatio support feedback communication as the theoretical background suggests.

Semi-structured interviews was selected for a research method because number of reasons. As the desired result is yet unknown, it makes sense to set the stage for the interview and let the discussion flow. According to Mason, the use of semi-structured interviews allow even unexpected themes to emerge. However, because the general themes of the interview are known beforehand, semi-structured interviews allow interviewer to ensure that the relevant contexts are brought into focus so that situated knowledge can be produced. (Mason, 2004)

The structure used for interviews can be found from appenfix A.

5.2 Finding interviewees

Hannotaatio is a publicly available tool, which can be used without registration. The ability to use the tool without registration is friendly for the users but it made contacting the users extremely difficult because the user contact information was not available. Providing a email address is only optional and thus the amount of email addresses in Hannotaatio's database is very limited. All of the users who had provided an email address and were identified by that address as an employee of a company closely working with software business were contacted and asked for an interview. Thus for example private persons were not contacted.

Hannotaatio's database of users' email addresses contains only email addresses to be used as a notification emails to the developers when a new feedback has been sent. Thus, the people who were contacted were all developers or other persons who were receiving the feedback via Hannotaatio, not sending it. For the research purposes, it would have been valuable to interview both roles of the feedback communication, feedback provider and feedback receiver. However, the contacted people were not very willing to the give contact information of their customers. Thus, only the feedback receivers were interviewed.

5.3 Preparing interviews

Before the interviews, a structure for the interviews was created. The purpose of this structure was to create a baseline, which was loosely followed. As the interviews

were semi-structured, the baseline structure left a lot of open space for new themes to arise.

A practice interview was conducted before the first recorded interview to test the content and the length of the interview structure. After the practice interview minor changes to the interview structure was made.

5.4 Conducting interviews

Nine people were interviewed in total. Eight of them were interviewed face-to-face and one was interviewed via email. Face-to-face was preferred because it allows interviewer to react on the response and possibly ask follow up questions. One of the nine interviewees was interviewed via email due to time restrictions and physical distance of the interviewee.

The interviews started with warm-up questions including basic information and job title of the interviewee and general description about the project where Hannotaatio was used. The middle section of the interview included more detailed questions about Hannotaatio as a feedback tool in a software project. The interviews ended with an open question where the interviewee was able to tell anything that she felt was missing.

All interviews were recorded. The language of the interviews was Finnish which is the native language for the interviewer and for all the interviewees.

5.5 Analyzing interviews

The data analysis process started in parallel with interviews. The analysis process loosely follows coding practice. As the interviews were recorded, the tapes were listened and the most important themes were written down.

The purpose of the analysis phase was to firstly find out a common themes shared between the interviews and secondly find out interesting viewpoints from individual interviewees.

6 Results of the semi-structured interviews

In this section, the analyzed results of the interviews are introduced. Common themes that were identified are put in their own subsection. In addition, quotes from interviews are presented in this section to give understanding what were the exact words interviewees used. Full transcriptions of the interviews are not available.

Nine people were interviewed in total. Eight were interviewed face-to-face in Helsinki area and one was interviewed via email due to scheduling issues. Each interview took from 45 minutes to one and a half hour depending on the talkativeness of the interviewee and the number of additional interesting topics emerged. All the interviews were recorded with the permission of the interviewee.

The interviewees were from five different companies. One company was Futurice, which is the company that sponsored the development of Hannotaatio feedback tool. Names of the other companies are not available. The size of the companies varied from small to large. In addition to Futurice, three other companies are also IT development and consultancy companies. One company is a telecommunication and ICT service company. Four of the interviewed people were from Futurice, five from the other companies.

Majority of the people interviewed are working in a company that offers IT development and consultancy services to other companies. In customer–supplier relationship this means that most of the interviewees were suppliers. Thus, they are the feedback receivers in supplier–customer relationship. The reason for this is, as noticed in previous chapters, that Hannotaatio's database contains only contact info of people who are receiving the notification email when a new feedback is sent to them. For the research purposes it would have been beneficial to interview more people from the sender side of the feedback communication.

The average age of interviewees was 29,5 years. Two of the interviewees were software developers, three user interface/user experience/concept designers, one head of internal IT, one service manager and one business manager. All of the interviewees were experienced with agile software development methodologies and were using agile methods in their daily work. Even though the job titles varied from software developer to designer and manager, all of the interviewees had a strong experience on software development and were in touch with agile software projects daily. All of the interviewees were working in companies where the work is done as project work. Some of the companies had their own products but even in those cases the work was done as project work.

The interviewees are not referred by their real name or their company. Instead, they are referred by a randomly generated[16] number. The linking between interviewees and numbers is not available.

The interviews were held in Finnish. Readers have to take into account that the quotes are translations from Finnish to English.

The structure of the email interview was different from the structure used in face-to-face interviews. The questions in the email interview were more detailed explained, because in case of misunderstanding the interviewer can not explain the

[16]http://www.random.org

questions further or correct the misunderstanding. The question 8 and its subquestions were omitted from the email interview. The reason for this is that due to the complex terms in this question it was noticed during the face-to-face interviews that this question required additional explanations so that the interviewees were able to properly understand the question. It would have been difficult to write sufficient but short textual description of the terms.

The interviews started with warmup questions to set the stage and to set the interviewee and interviewer to the right mood. At the beginning of the interviews the interviewees were asked to briefly tell about their company and what does the company do. After that they were asked to briefly describe the most recent or two most recent projects they have been involved. This way the interviewee had to recap the project in his or her mind and so mentally prepare to the following questions. To get answers from interviewees own perspective and not from the general perspective the following questions referred to the two last projects of the interviewee. For example, instead of asking "how feedback is given in your company's project" the interviewees were asked to describe how feedback was given in their lastest project.

6.1 Feedback communication methods and media

Before the interview proceeded to Hannotaatio and the different properties of a feedback communication media, the interviewees were asked to describe how they have conducted feedback communication in their two most recent projects.

> *(IW 8): We are daily in touch with the people who are going to use the end product, in other words we are in close contact with the customer. We can just go and ask the customer how this [implemented feature] worked. [...] [The main part of the communication] happens through one-to-one communication, whether it is via email or telephone.*

> *(IW 5): The process [for feedback communication] was pretty clear. It was the same process I have used for all the projects I've been managing. Since it has worked previously, why change? Particularly we try to be as close to the customer as possbile.*

When interviewees were asked what are the communication tools and methods to get feedback it was interesting to hear how little companies put effort on finding new and innovative ways for feedback communication. All answers were more or less similar stating that the feedback communication is done "the usual way". When I asked them to list communication media they use, the most commonly used media were face-to-face, telephone, email and Skype chat[17]. Obviously, also Hannotaatio was mentioned, because it was known in advance that all the interviewed people have used it. When I asked reasons for choosing exactly those communication media, the answers were easiness of use, familiarity and face-to-face being the most efficient according to the interviewees.

[17]http://www.skype.com/en/

All of the interviewees emphasized the importance of intense communication and most saw face-to-face being the superrior communication method. Physical proximity to the customer was also highly valued.

6.2 Most common problems in feedback communication

> *(IW 5): If we take a look at the customers that I have had, I assume the biggest problem is that they have very little time for the project in addition to the project meetings. This means that they are not the ones who use the application in the evenings [at home]. Very seldom the customer actually uses the service she is building. And when she is not using it, it is extremely difficult to get the customer to give the feedback.*

Some of the interviewees felt that they did not get enough feedback from the customer. On the other hand, some got enough feedback, but they felt that they were unable to utilize it. In both cases, there were problems in the ways how teams collect and utilize feedback.

When the reasons were discussed, it came up that the reasons were mostly social or process related. The feedback communication media was very seldom seen as the reason for lack of feedback. In some projects, there were not a dedicated person or persons who would have been responsible for giving feedback or the person responsible for giving feedback did not have enough time for the project.

On the other hand, some projects did not have a dedicated person to collect the feedback, analyze it and act accordingly. However, it is worth noting that this particular issue arose in an internal development project, where there were no official organizational boundaries crossing supplier–customer relationship. This might very well be the reason why the utilizing of the feedback was not in the main focus of the project. Due to the fact that the supplier–customer relationship was missing from the internal project, its feedback process was not taken as seriously as in supplier–customer projects.

In some projects the review and feedback processes were not in place. For example, while code was regularly reviewed by peer programmers, the application user interface design was not reviewed at all, before it went to production. This lead the user interface to be lower quality than the customer expected.

It is important to notice that the feedback tool is just one part of the successful feedback communication. If the project is missing people who are responsible for giving and receiving feedback, based on the interviews, it is likely that the team will eventually suffer from a lack of feedback. By just changing the feedback communication media or modifying the existing media rarely helps the case. No matter how efficient the feedback communication tool is, it does not help much if the tool is not used.

However, even though the feedback tool can not solve all the issues which cause the lack of feedback, it came up that there are some things the feedback media could do in order to promote the feedback communication. One of the interviewees suggested that the feedback tool could promote itself, for example in the case of

Hannotaatio, a small popup could be shown to encourage and remind the user to use the feedback tool.

Another interviewee pointed out that feedback is something that the feedback receiver has to proactively ask. Feedback is rarely given, if everything is going somewhat good, nothing major is broken and no one is asking for feedback. He explained that one of the reasons for this is that giving feedback does not fit well to the use flow of the application. When a user uses an application, she has some goals she wants to accomplish, for example add a new item to an e-commerce store application or set a due date to an item in ToDo application. Giving feedback does not fit well to these goals, instead, it is actually an extra effort for the user. To address this issue, he and one another interviewee suggested to link the feedback tool to the use session. In practice this means that after the user has tried the application and is closing the browser window or otherwise leaving the application, the feedback tool could at that moment ask feedback from the user. This behaviour would not interfere the normal use of the application but it would encourage to give feedback when the time is appropriate.

6.3 Different communication media for different abstraction level

> *(IW 3): One thing that people don't always get is that what is the proper level of abstraction for the discussion at the moment. For example, when we are in the very beginning of a design process of a new set of features, some people dive in to the color of a button. When the conversation goes to that level it becomes very slow and doesn't progress, instead, we're talking about wrong subject at the wrong time. [...] On the other hand, it can be that certain people always think on a very detail level, where as others always think on a high abstraction level. For them it's not easy to move from level to level. There are also people how are able change the abstraction level as the situation demands.*

During the interviews it was also pointed out by two interviewees that the abstraction level of the discussion has a major effect on the selection of communication media.

In software development, there are multiple work phases, which all have a different abstraction level. As an example, the project often starts from high-level requirements gathering. The concept designer will then design a high-level concept of the application. The concept can be documented in a static document or it can be even quickly implemented in a prototype. The next design phase is to design the application layout and the graphical elements. After that, the actually software is developed.

Feedback about the concept design, the user interface (UI) design and the actual software product have very a different level of abstraction. Feedback about the concept design or a proof-of-concept prototype should be kept in a high abstraction

level. If the feedback goes too much into the implementation details of the application or into the details of the UI graphics, the value of the feedback conversation decreases. On the other hand, when the feedback is given acout the UI design or actual working piece of software, the feedback can be given with much higher detail level.

The level of abstraction highly affects to the selection of communication media. For example, as one of the interviewees mentioned, Hannotaatio does not support well feedback of a high level of abstraction. Instead it even directs the user to give feedback about the details, which are not important during certain design phase of the application. He said that the tools in Hannotaatio, namely arrow, drives people to point details instead of commenting the concept.

6.4 Feedback as a weapon

> *(IW 8): I have quite a lot of experience on situations, where wrong people use a one single user feedback comment as tool and evicende to drive their own opinions in the project, even though in reality, the feedback is from one single user and from one single situation. It should not weight much in a desicion making process.*

Conflicts happen in supplier–customer relationship. The conflicts also affect on feedback communication. According to the interviews it seems that feedback is often used as a weapon (or a shield, depending on your position) in a conflicting situations. For example, from the quote above it can be seen that sometimes one single feedback is given way too much weight, if it supports one's own opinions.

Generally it seems that there are often situations where previously given feedback is referred in a communication between the supplier and the customer. The need to go back to the feedback given earlier was one of the main reasons why almost all interviewees mentioned that feedback tool has to support ability to reread and reprocess the previously given feedback. Some interviewees mentioned that the ability to reread the feedback is essential in a conflicting situations to support the reasoning behind a previously made change. For example, in a conflict situation where customer is not satisfied about something the development team has done, the previously given feedback, where customer says he wants the development to do this and that, is used by the development team to articulate why they have done the change the way they have.

6.5 Storing the feedback

> *(IW 3): It may be so that one remembers better the feedback that is similar to your own opinions where as some other feedback [contradictory to your own opinions] you forget easily. Thus, it might be good that one could go back to the previous feedback.*

In addition to conflicting situations, there were other reasons why storing the feedback for later use was seen important by almost all interviewees. As seen from

the above quote, one interviewee realized during the interview that in his previous project he might have been ignoring some feedback received from the customer in face-to-face meetings. This was unintentional, but it might have happened subconsciously because the feedback was contradictory to his own opinions. This led to problems because customer expected the feedback to be taken into account. According to the interviewee, the possibility to store the feedback and review it might help to address the issue.

Another interviewee mentioned that it is extremely important to be able to store and reprocess the feedback because one interprets the received message differently based on your mood and energy level. For example, when one is angry or tired, the feedback might be interpret very differently than it would be in other situations. The interviewee mentioned that if one notices the low energy level when the feedback is received, it might be good idea to reprocess the feedback after couple hours with higher energy level and better mood.

6.6 More natural or more awkward?

Kock (2005) argues that "modern humans' brains are not optimally adapted for current e-communication technologies because these technologies often suppress too many of the elements found in face-to-face communication". In this case, Hannotaatio is not an exception to the other available e-communication technologies. In fact, Hannotaatio does not support transformation of any physical cues, such as body language or vocal tone. According to media naturalness theory (MNT), this leads to an increase in cognitive effort.

Interviewees were asked about new features that could be implemented in Hannotaatio and which would increase the capability of transmitting the elements that are found in face-to-face communication. The new features were: ability to record screen capture video, ability to record screen capture video including speech and ability to record screen capture video, include speech and include webcam recording of the person who is giving the feedback, thus transmitting also the body language and vocal tone.

The interviewees' answers did not fully support MNT. According to MNT and Kock's later research, the burden of the e-communication use in on the sender side (Kock, 2007). Increasing the naturalness of communication media should thus decrease the cognitive effort of the message sender and make the communication media more appealing to the sender. However, most of the interviewees answered that they would not be too keen on recording their face while giving feedback because it would feel awkward. In addition, recording speech was generally seen as a positive feature, but some negative comments were also raised. One interviewee said that recording speech would work well, but not here in Finland because "we are so shy". Another said that recording speech is generally good but may not be convenient for a message sender to record speech in a quite office environment.

6.7 Social perspective

> *(IW 2): It was easy to sell the idea [to start using Hannotaatio] to the customer because I knew the customer very well and they trusted that if I suggest something, it is a wise suggestion.*

> *(IW 5): Interviewer: Why Hannotaatio was used so actively in your projects?*
> *Interviewee: Because I told them to use it. That was probably the biggest reason.*

Even thought the social perspective of the communication media is not in the scope of this thesis the interviews revealed that the social perspective has a lot of effect on communication media selection. As pointed out by one of the interviewees, at the end of the day, software business is people business. People do agreements and suppliers have to keep up good customer service level. IT consultant companies are not only delivering software, they are also delivering customer service. This highly effect on the communication media selection.

In the interviewees it was clear that in some projects Hannotaatio was much more actively used than in others. The reasons why Hannotaatio was not actively used in some projects were often people or process related issue, for example not having proper process for receiving feedback or not having a person responsible for giving feedback. On the other hand, the reasons why Hannotaatio was actively used in other project were also people related. As pointed out by IW2, the customer started to use Hanotaatio, because they knew IW2 well and trusted him. IW5 explained that he more or less told customer to use Hannotaatio.

The social pressure has also been noted in previous research. El-Shinnawy and Markus (1997) found out that in some organizations email was used widely, because managers told subordinates to use email.

6.8 Scoping the feedback to the part that changed

Some interviewees pointed out that even though more feedback is usually better, sometimes the feedback is received about subjects that are not important at the time. They were in need for a communication tool, which would allow scoping the subject under discussion. For example, one interviewee noted that if the abstraction level of the discussion goes in too detailed level too early, feedback is given about wrong subject.

Another interviewee pointed out that customers may want to give feedback about features that were implemented and accepted in much earlier releases. The interviewee suggested a new feature to commnication tool, which could help scoping out the unchanged parts. Usually, when a new release is delivered to the customer, the development team is interested in hearing feedback about the newest changes. The feedback tool could help to just show the changed part of the application to the customer.

6.9 Important properties for feedback communication media

> *(IW 2): It's always nice to discuss face-to-face but it also brings some issues from the software development point-of-view. If a developmer is in the coding "flow", the interuption leads 20min break [before the coder is productive again]. One should always consider if the interuption caused by face-to-face discussion is worth the time.*

The communication media theories list a wide range of properties that affects medium's ability to support specific communication task. During the interview the interviewees were asked how much they value the given property in the communication media. The list of properties were collected from media richness theory (MRT), media synchronicity theory (MST) and media fitness theory (MFT). The list included the following properties:

- immediate feedback (MRT, MST, MFT)
- personality (MRT)
- rich symbol set (MRT, MST, MFT)
- natural symbol set (MRT, MST, MFT)
- parallelism (MST)
- rehearsability (MST)
- reprocessability (MST)
- security (MFT)
- sharing (MFT)
- retrieval (MFT)
- multiparty (MFT)
- available time (MFT)
- available location (MFT)
- price (MFT)

In this part of the interview interesting results arise. In the beginning of the interview, when interviewees were asked to describe how they have conducted feedback communication in their previous projects, almost all interviewees said face-to-face communication is used widely. They also emphasized the importance of face-to-face communication and being close to the customer.

There seems to be a contradiction between the communication media interviewees value in the real life and in theory. At the beginning of the interview the

interviewees said they value face-to-face communication and stated that it is the best and most effective communication media. However, later during the interview when they were asked their opinion about different properties in communication tools, the most valued properties were not supported by face-to-face communication. For example, the two most valued properties for communication media were time independent availablity and location independent availability. Time independent availability stands for ability to give feedback dispite the current time of the day, for example, not only during the office hours. The same way, location independent availability stands for availabilty to give feedback from any physical location, for example, not only from the office.

Face-to-face communication is synchnorous and thus it does not support time and location independent availability the same way as asynchronous e-communication media, such as email. In fact, face-to-face communication requires all participants to be at the same location at the same time. According to interviews, this is often impossible in software projects where the customer and supplier may not be colocated, not even in the same country.

In addition to time and location independent availability, also other properties that are not available in face-to-face communication were valued. After time and location independent availability the next most valued properties were immediate feedback, natural symbol set, reprocessability and sharing abilities. Face-to-face communication supports immediate feedback and it has the most natural symbol set. However, it does not allow the feedback receiver to reprocess the feedback after it has been given. On the otherhand, sharing face-to-face feedback can not happen without information loss (Higa and Gu, 2007). Sharing the face-to-face message to another person who was not part in the original communication situation loses some parts of the message that the perseon who is sharing the message is not able to convey, for example, the exact vocal tone and body language of the original message sender.

Most e-communication media support reprocessability and sharing well. The message sent via e-communication tool can be easily recorded and stored for later use whether the media is instant messaging (IM) system or email. The ability to record and store the feedback sent earlier allows reprocessing of the message and sharing it to others without any information loss. However, e-communication tools, such as email, have problems to support immediate feedback and natural symbol set. For example, written text is not natural compared to face-to-face communication and even though sending and transmitting the email happens instantly, getting the response to the message might take hours, which lowers the immediancy of the feedback. There are e-communication tools which address these issues, for example video conferencing system, but being synchronous communication tool, video conferencing system does not allow time independent messaging.

It has to be taken into account that some of the properties are in contradiction with each other. For example, immediate feedback and time independent availability are contradictory properties. If another communication participant is located for example other side of the globe, due to the time difference it is just impossible to give feedback at any time and expect to get immediate feedback. However,

contradictory or not, both of the properties were highly desired by the interviewees.

Clearly, with the current available technology, there are no communication media that would support both, immediate feedback and time independent availability. At the moment the communication media users have to do a trade-off between these two properties. With the help of synchronous media, the responses are immediate but not timely independent. On the otherhand, with the help of asynchronous media, feedback can be given at any time but the responses are not immediate.

7 Conclusions

In this chapter, I answer the research question set in section 1.2: *what the are properties that make a communication tool effective for giving and receiving feedback in a agile software projects?* The research question is answered based on the interview results presented in the previous chapter.

The interviews covered three themes that were predefined in the interview structure: current feedback communication methods and media, the properties that are important for the feedback communication media and the evaluation of Hannotaatio as a feedback communication tool in real-life projects. In addition, during the semi-structured interview, seven new interesting themes arose: common problems in feedback communication, the need for different communication media for different abstraction level, the use of user feedback as a weapon, the importance of storing the feedback, the awkwardness of the more natural communication media, the importance of social aspects and the importance of scoping the feedback down to the part of the application that has changed.

7.1 Communication media properties

Face-to-face communication has been argued to be the "best" communication media according to many of the communication theories studied in this thesis. Media richness theory (MRT) states that face-to-face communication is the "richest" available communication method. Media naturalness theory (MNT) positions face-to-face communication in the middle of media naturalness continuum arguing that all other media are either more or less natural. The media more or less natural than face-to-face communication increases the cognitive effort of the media use. In addition, interviewees valued face-to-face communication and stated it to be the best and the most efficient communication media. Immediate feedback in face-to-face discussion was the most important property that made the face-to-face communication efficient according to the interviewees.

However, interviewees admitted that face-to-face communication is not always possible due to its limitations. Face-to-face communication requires shared time and shared physical location. Interviewees agreed that these are the most important reasons why other communication media are often used instead of face-to-face communication.

The six most valued properties in a communication media that is used for feedback communication were according to interviewees: time independent availability, location independent availability, immediate feedback, natural symbol set, ability to reprocess the feedback and ability to share the feedback to people who were not present in the original feedback communication session.

From the six most valued properties it can be seen that face-to-face communication supports poorly other properties than immediate feedback and natural symbol set. Face-to-face communication requires communication participants to be at the same physical location at the same time, thus it does not support time and location indepented availability. In addition, if the face-to-face conversation is not recorded,

the receiver of the message does not have any means to reprocess the message after the communication session. Also, if the converstation is not recorded, the sharing of the message loses information (Higa and Gu, 2007). Because of this, I argue that even though face-to-face communication was seen as the most efficient communication media by the interviewees, it does not fulfill all the needs for feedback communication media. Thus, there is a need to design and implement new and even better communication media that would surpass traditional face-to-face communication.

7.2 Evaluation of Hannotaatio in a real-life software projects

A prototype of a feedback communication tool, Hannotaatio, was implemented to answer the need for a better feedback communication media. Compared to email which is, according to the interviewees, one of the most used tool for feedback, Hannotaatio adds an important visual element to the communication media. According to the media theories and interviewees, the added visual element makes Hannotaatio more natural than email whose symbol set is constructed from written text.

I argue that Hannotaatio supports well five of the six most valued feedback tool property. This makes it a noticeable feedback communication medium. However, Hannotaatio supports poorly immediate feedback, which was one of the most valued property and the reason why face-to-face communication was ranked the most efficient media by the interviewees.

If we review the properties that were, according to the interviewees, the most important for a feedback tool, it can be seen that Hannotaatio supports well the properties the interviewees valued the most. The most highly valued properties were time and location independent availability, immediate feedback, natural symbol set, reprocessability and sharing abilities.

Hannotaatio supports extremely well reprocessability, sharing, available time and available location. In addition to these properties, a lot of development work has been done to support natural symbol set and immediate feedback.

The visual symbol set used in Hannotaatio is natural and it fits well to the purpose of giving feedback about software product. According to many interviewees, it is much easier to take the screenshot using Hannotaatio than explaining the problem in email. In fact, I argued that in some cases the symbol set used in Hannotaatio is more natural than the symbol set available in face-to-face communication. If the face-to-face communication happens in an environment where neither printed screenshots or a computer with the software under the feedback conversation is running, the communication parties have to verbally explain different parts of the application. This task can be troublesome. Much easier way to discribe a problem in a software is just point the location from the screenshot. This idea is also supported by media synchronicity theory (MST). Dennis et al. (2008) gives an example of this: "Imagine that you are trying to describe how to perform a physical activity on a Web site. A verbal description alone is likely to be less effective than a visual demonstration and a verbal description or a series of annotated screen shots with a written description."

Immediate feedback was valued by interviewees but is not well supported in Hannotaatio. The immediacy of the feedback in Hannotaatio is the same as in email, due to the fact that the notification to the feedback receiver is sent via email. Because of this, after sending the feedback the conversation continues via email and email does not support immediate feedback.

However, interviewees gave some improvement ideas that could address the issue of low immediacy of the feedback. One idea was to add an instant messaging pane to Hannotaatio. In the messaging pane, the feedback receiver could ask follow up questionss from the sender. This way the immediacy of feedback would raise to the next level, from email level to IM level. The immediacy level of IM is not as high as in face-to-face communication, but at least it is better than in email. In addition, the messaging pane would allow communication to continue in Hannotaatio, not in another communication media, namely email. This would supposedly lower the effort of communication media use, when users of the media do not need use two media at the same time.

In conclusion, it seems that from the six most valued communication media properties, Hannotaatio is able to implement five of them. The table 12 lists the properties and Hannotaatio's support in a scale of low, medium, high.

Table 12: Hannotaatio's support for properties valued by the interviewees

Communication media property	**Hannotaatio's support**
Time independent availability	High
Location independent availability	High
Immediate feedback	Low
Natural symbol set	High
Reprocessability	High
Sharing	High

According to interviewees, Hannotaatio seems to be effective communication tool. No one said that it would surpass face-to-face communication, but many said Hannotaatio is "the second best option" if face-to-face is not possible.

In order to improve Hannotaatio to surpass face-to-face communication, the problem of synchronous versus asynchronous communication has to be solved. Interviewees valued both synchronous properties such as immediate feedback and asynchronous properties such as time and location independent availability. Obviously, with the current technology, it is impossible to build a communication tool that would be both synchronous, providing support for immediate feedback, and asynchronous, providing ability to send the message when ever, where ever. Communication tool that would be both synchronous and asynchronous at the same time should have some kind of time travelling capabilities.

7.3 Other emerged themes

The other themes that emerged from the interviews can help the development of new feedback tools. For example, in some cases there were problems finding a correct level of abstraction to the feedback conversation. The developers of new feedback tools should take this issue into account. How could the feedback tool help communication participants to find the correct abstraction level to the discussion? On the other hand, how could the feedback tool prevent people from using one single user feedback as a weapon to drive through their own agendas?

One might thing that the problems discribed above are social issues between people and irrelevant for the technical implementation of the feedback tool. During the interviews it was also noticed that problems with people and processes cause a lot of other problems in feedback communication. It might be true that part of the issue lies on the people using the feedback tool, but it does not mean that the feedback tool can not help the people to solve these issues.

The results of this thesis provide insight about feedback tool properties for new feedback tool developers, even though the problem of combining asynchronous and synchronous communication remains unsolved. In addition to the six most valued properties identified in the interviews, various new and concrete improvement suggestions emerged.

One of the improvement ideas that arise from the interviews was ability scope the feedback conversation to include the part of the application that has changed. According to the interviewees, too often the feedback conversation loses its focus. Instead of giving feedback about the new features, customers tend to give feedback about old features that have been implemented and accepted long ago. On the other hand, instead of giving feedback, customers may start to innovate new ideas and forget to give the feedback about the features the team has implemented in the previous iteration.

Another improvement idea from the interviews was to connect the feedback tool to the user's application use flow and proactively ask user to give feedback. According to the interviewees, the feedback tool should be smart enough to analyze users use flow and ask feedback after the user has ended the use session. With the help of this feature, the feedback giving would be seamlessly tied to the use flow and thus make the effort of giving feedback smaller to the user.

8 Discussion

In this section, the research internal and external validity is discussed. The internal validity chapter discusses whether the methods used in this study were appropriate and whether they were used correctly. The external validity chapter discusses if the results can be generalized.

8.1 Internal validity

During the research I identified several issues that can threaten the internal validity. In this chapter the issues are discussed.

Most of the interviewees were the ones who introduced Hannotaatio to their project team. Because they were already willing to try out Hannotaatio and its suitability, they had to have some kind of special interest towards new communication media. This admittedly affected positively to their view on Hannotaatio. For the research purposes it would have made sense to select random people from software projects who did not have previous experience on Hannotaatio.

When it was asked how did the interviewed people found the Hannotaatio tool in the first place, it became evident that most of them had some kind of connection to Futurice and had heard about Hannotaatio through that connection. For example, some of them had worked with Futurice previously and heard about Hannotaatio that way. Others had friend who are working for Futurice and heard about Hannotaatio from them. Only one of the interviewed people had discovered Hannotaatio randomly. This admitedly may have affected the results of the interviews.

Most of the interviewed people were not using Hannotaatio actively at the moment of interviewing. Instead, they had been using it in the past. Only two of nine were using Hannotaatio in the project in which they are currently working. According to Silverman (2009) this is not an optimal situation. Instead, one should conduct an interview so that the interviewed people do not have to rely on their memories when answering the questions. Answering to the interview questions based on one's memory may have an effect on the results.

In the interviews a contradictory results arose. Interviewees said that they value properties in a communication media that are not available in face-to-face communication, but still they said that they prefer face-to-face communication. In an interview research, the interviewer has to trust the interviewee. However, when a contradictory results like this arise, the researcher naturally starts to thing whether the interviewees said one thing but in reality did another. To overcome this issue, this research would have benefited from case study and observational research. Instead of asking interviewees to describe the feedback methods they have used, the observation of a real feedback communication would have given more presice results.

Majority of the interviewees knew that the interviewer was heavily involved in the development of Hannotaatio. There is a possibility that this would have affected on their answers. They may have wanted to please the interviewer and thus give too positive answers about the Hannotaatio prototype. However, as an interviewer, I tried to be as neutral as possible.

8.2 External validity

It has to be noted that most of the interviewed people were software suppliers, not customers. This might have an effect to the results of the interviews. For example, most of the interviewees emphasized face-to-face communication. The reasons varied, but for most of them the social reasons were important. The supplier wanted to communicate to customer that they are being taken care of and their feedback will be taken care of. They wanted to keep the level of customer support high. This was one of the reasons why face-to-face communication was valued.

For suppliers the case might be opposite. As noted in interviews, the customers may have very limited amount of time. Thus, when they give feedback they want to do it fast, they expect it to be understood, the follow up actions to be conducted. For this purposes, they may want to use fast communication methods such as email instead of face-to-face communication.

When people were interviewed in this study, only a limited coverage of different companies was accieved. Due to the limited number of Hannotaatio users, many of the interviewed people were from the same company. Almost half of the interviewees, four out of nine, were from Futurice, the company that supported the development of Hannotaatio. The other four companies were quite alike Futurice. They had much the same ways of working, similar agile processes and open and modern company culture. Only one of the companies was significanly bigger than the rest.

Due to the fact that the interviewed people were from companies that much resemble each other, it can be argued that the results may not be applicable for different kind of software companies. For example, many companies work in a globally distributed environment. Parts of the development team can be on otherside of the globe. Because all the interviewed companies had colocated teams, it is unsure whether the results hold in a distributed software development context.

8.3 Further research

The interviewers gave contradictory answers about the preferred way to communication feedback and the properties that are important for feedback communication tool. Even though they emphasized importance of physical proximity to the customer and valued face-to-face communication, they also appreciated the properties in Hannotaatio that are not available in face-to-face communication, such as reprocessability, ability to share and give feedback when ever, where ever. Further research could be conducted about the combination of the properties in face-to-face communication and e-communication tools. It would be also interesting to push the limits of human knowledge and try to solve the dilemma, how to be able to give feedback when ever and receive the feedback immediately.

References

Kent Beck and Cynthia Andres. *Extreme programming explained: embrace change*. Addison-Wesley Professional, 2004.

Kent Beck, Mike Beedle, Arie van Bennekum, Alistair Cockburn, Ward Cunningham, Martin Fowler, James Grenning, Jim Highsmith, Andrew Hunt, Ron Jeffries, Jon Kern, Brian Marick, Robert C. Martin, Steve Mellor, Ken Schwaber, Jeff Sutherland, and Dave Thomas. Manifesto for agile software development, 2001. URL `http://agilemanifesto.org/`.

John R. Carlson and Robert W. Zmud. Channel expansion theory: A dynamic view of medial and information richness perceptions. In *Academy of Management Best Papers Proceedings*, pages 280 – 284, 1994. URL `http://search.ebscohost.com.libproxy.aalto.fi/login.aspx?direct=true&db=bth&AN=10344817&site=ehost-live`.

John R. Carlson and Robert W. Zmud. Channel expansion theory and the experiential nature of media richness perceptions. *The Academy of Management Journal*, 42(2):pp. 153–170, 1999. ISSN 00014273. URL `http://www.jstor.org/stable/257090`.

Alistair Cockburn and Jim Highsmith. Agile software development, the people factor. *Computer*, 34(11):131–133, 2001.

Richard L. Daft and Robert H. Lengel. Organizational information requirements, media richness and structural design. *Management Science*, 32(5):pp. 554–571, 1986. ISSN 00251909. URL `http://www.jstor.org/stable/2631846`.

Richard L. Daft, Robert H. Lengel, and Linda Klebe Trevino. Message equivocality, media selection, and manager performance: Implications for information systems. *MIS Quarterly*, 11(3):pp. 355–366, 1987. ISSN 02767783. URL `http://www.jstor.org/stable/248682`.

Alan R. Dennis and Susan T. Kinney. Testing media richness theory in the new media: the effects of cues, feedback, and task equivocality. *Info. Sys. Research*, 9(3):256–274, March 1998. ISSN 1526-5536. doi: 10.1287/isre.9.3.256. URL `http://dx.doi.org/10.1287/isre.9.3.256`.

Alan R. Dennis, Robert M. Fuller, and Joseph S. Valacich. Media, tasks, and communication processes: a theory of media synchronicity. *MIS Q.*, 32(3):575–600, September 2008. ISSN 0276-7783. URL `http://dl.acm.org/citation.cfm?id=2017388.2017395`.

A.R. Dennis and J.S. Valacich. Rethinking media richness: towards a theory of media synchronicity. In *System Sciences, 1999. HICSS-32. Proceedings of the 32nd Annual Hawaii International Conference on*, volume Track1, page 10 pp., 1999. doi: 10.1109/HICSS.1999.772701.

Darleen M. DeRosa, Donald A. Hantula, Ned Kock, and John D'Arcy. Trust and leadership in virtual teamwork: A media naturalness perspective. *Human Resource Management*, 43(2-3):219–232, 2004. ISSN 1099-050X. doi: 10.1002/hrm.20016. URL http://dx.doi.org/10.1002/hrm.20016.

Maha El-Shinnawy and M. Lynne Markus. The poverty of media richness theory: explaining people's choice of electronic mail vs. voice mail. *International Journal of Human-Computer Studies*, 46(4):443 – 467, 1997. ISSN 1071-5819. doi: 10.1006/ijhc.1996.0099. URL http://www.sciencedirect.com/science/article/pii/S1071581996900993.

Janet Fulk, Charles W. Steinfield, Joseph Schitz, and J. Gerard Power. A social information processing model of media use in organizations. *Communication Research*, 14(5):529–552, 1987. doi: 10.1177/009365087014005005. URL http://crx.sagepub.com/content/14/5/529.abstract.

Futurice. Hannotaatio, 2010. URL https://hannotaatio.futurice.com/.

A. Graveline, C. Geisler, and M. Danchak. Teaming together apart: emergent patterns of media use in collaboration at a distance. In *Professional Communication Conference, 2000. Proceedings of 2000 Joint IEEE International and 18th Annual Conference on Computer Documentation (IPCC/SIGDOC 2000)*, pages 381 –393, 2000. doi: 10.1109/IPCC.2000.887296.

R. Gu, K. Higa, and D.R. Moodie. A study on communication media selection: Comparing the effectiveness of the media richness, social influence, and media fitness. *Journal of Service Science and Management*, 4(3):291–299, 2011.

E. Gummesson. *Qualitative Methods in Management Research*. SAGE Publications, 1999. ISBN 9780761920137. URL http://books.google.fi/books?id=11axQgAACAAJ.

Alan R Hevner, Salvatore T March, Jinsoo Park, and Sudha Ram. Design science in information systems research. *MIS quarterly*, 28(1):75–105, 2004.

K. Higa and R. Gu. Communication media selection for remote work: Towards a theory of media fitness. *The Journal of E-working*, 1:45–68, 2007.

S. Josefsson. The Base16, Base32, and Base64 Data Encodings. RFC 4648 (Proposed Standard), October 2006. URL http://www.ietf.org/rfc/rfc4648.txt.

N. Kock. The psychobiological model: Towards a new theory of computer-mediated communication based on darwinian evolution. *Organization Science*, 15(3):327–348, 2004.

N. Kock. Media richness or media naturalness? the evolution of our biological communication apparatus and its influence on our behavior toward e-communication tools. *Professional Communication, IEEE Transactions on*, 48(2):117 – 130, June 2005. ISSN 0361-1434. doi: 10.1109/TPC.2005.849649.

Ned Kock. Media naturalness and compensatory encoding: The burden of electronic media obstacles is on senders. *Decision Support Systems*, 44(1):175 – 187, 2007. ISSN 0167-9236. doi: 10.1016/j.dss.2007.03.011. URL http://www.sciencedirect.com/science/article/pii/S0167923607000632.

Kirsi Korhonen. Evaluating the impact of an agile transformation: a longitudinal case study in a distributed context. *Software Quality Journal*, pages 1–26, 2012. ISSN 0963-9314. doi: 10.1007/s11219-012-9189-4. URL http://dx.doi.org/10.1007/s11219-012-9189-4.

M. Korkala, P. Abrahamsson, and P. Kyllonen. A case study on the impact of customer communication on defects in agile software development. In *Agile Conference, 2006*, pages 11 pp. –88, july 2006. doi: 10.1109/AGILE.2006.1.

Maarit Laanti, Outi Salo, and Pekka Abrahamsson. Agile methods rapidly replacing traditional methods at nokia: A survey of opinions on agile transformation. *Information and Software Technology*, 53(3):276–290, 2011.

C. Larman. *Agile & Iterative Development: A Manager's Guide*. The Agile Software Development Series. Addison-Wesley, 2004. ISBN 9780131111554. URL http://www.google.fi/books?id=76rnV5Exs50C.

Jennifer Mason. *Semistructured Interview*, pages 1021–1022. Sage Publications, Inc., 0 edition, 2004. URL http://srmo.sagepub.com/view/the-sage-encyclopedia-of-social-science-research-methods/n909.xml.

K. Nakamura, Y. Masuda, and Y. Kiyokane. Roles of communication media in telework environments. In *System Sciences, 1995. Proceedings of the Twenty-Eighth Hawaii International Conference on*, volume 4, pages 446 –455 vol.4, jan 1995. doi: 10.1109/HICSS.1995.375704.

Jay F Nunamaker Jr and Minder Chen. Systems development in information systems research. In *System Sciences, 1990., Proceedings of the Twenty-Third Annual Hawaii International Conference on*, volume 3, pages 631–640. IEEE, 1990.

Roman Pichler. *Agile Product Management with Scrum: Creating Products that Customers Love*. Addison-Wesley, 2010. ISBN 978-0-321-60578-8.

Travis Reitter. Let's push things forward, February 2013. URL http://treitter.livejournal.com/14871.html.

Ronald E. Rice and Gail Love. Electronic emotion: Socioemotional content in a computer-mediated communication network. *Communication Research*, 14(1): 85–108, 1987. doi: 10.1177/009365087014001005. URL http://crx.sagepub.com/content/14/1/85.abstract.

Winston W Royce. Managing the development of large software systems. In *proceedings of IEEE WESCON*, volume 26. Los Angeles, 1970.

Joseph Schmitz and Janet Fulk. Organizational colleagues, media richness, and electronic mail: A test of the social influence model of technology use. *Communication Research*, 18(4):487–523, 1991. doi: 10.1177/009365091018004003. URL http://crx.sagepub.com/content/18/4/487.abstract.

K. Schwaber. *Agile Project Management with Scrum*. Microsoft Press, 2009. ISBN 9780735637900. URL http://books.google.fi/books?id=RpYX01XVMksC.

James Shore and Shane Warden. *The art of agile development*. O'Reilly Media, Incorporated, 2007.

D. Silverman. *Doing Qualitative Research*. SAGE Publications, 2009. ISBN 9781848600348. URL http://books.google.fi/books?id=J1ZFVvPPPpsC.

Andy Singleton. Seven things i hate about agile, August 2012. URL http://blog.assembla.com/assemblablog/tabid/12618/bid/87899/Seven-Things-I-Hate-About-Agile.aspx.

Bruce Tyson. How secure is email, 2011. URL http://www.brighthub.com/computing/enterprise-security/articles/105341.aspx.

Wikipedia. Software as a service, a. URL http://en.wikipedia.org/wiki/Software_as_a_service.

Wikipedia. User story, b. URL http://en.wikipedia.org/wiki/User_story.

A Interview Questions

Preface: I am doing this interview for my Master's Thesis at Aalto University Information Networks degree. The purpose of the interview is to study the use of the Hannotaatio feedback tool and feedback communication in software projects in general. The interview will be recorded.

1. Job title and age? Am I allowed to use this information in my thesis?
2. Which company are you working for? Am I alloed to use this information in my thesis?
3. Describe briefly what do you do, and what your company does?
4. How do you do feedback communication in customer–supplier relationship?
5. What the are the elements that define how and with what tools the feedback is communicated?
6. Describe briefly the project in which you used Hannotaatio?
7. Did the introduction of Hannotaatio change the feedback communication somehow?
   (a) How much feedback did you receive via Hannotaatio?
   (b) Why so much / why so little?
8. How much do you value the following properties in a feedback communication tool?
   (a) The communication channel has to be continue to be open after the receiving / sending the feedback in order to allow for example further questions? *Immediate feedback (MRT, MST, MFT)*
   (b) Personality, that is, you can see and feel the personality of the feedback sender *Personality (MRT)*
   (c) Ability to express the feedback in numerous ways or symbols, such as text, image, voice, video, gestures, voice tone? *Number of cues / language variety (MRT, MST, MFT)*
   (d) Ability to express the feedback in ways that are natural for humen (for example, picture is more natural than text, video is more natural than voice) *Natural symbol set* (MRT, MST, MFT)
   (e) Ability to be in multiple comminucation sessions simultaneously (for example, face-to-face communication allows only one simultaneous session where as instant messaging (IM) allows multiple chat rooms open at the same time) *Parallelism (MST)*
   (f) The ability to change, modify and finalize the feedback message before sending it *Rehearsability (MST)*

(g) Ability to return to the feedback message and reinterpret it *Reprocessability (MST)*

(h) Security, the data is exposed only with the communication participants *Security (MFT)*

(i) Ability to easily share the message to people how were not present in the original communication session *Sharing (MFT)*

(j) Ability to index the feedback message for searching purposes *Retrieving (MFT)*

(k) Ability to use by multiple users at the same time (similar to Google Docs) *Multiparty (MFT)*

(l) Ability to give feedback regardless of time *Availability time (MFT)*

(m) Ability to give feedback regardless of location *Availability location (MFT)*

(n) The cost of the communication tool *Cost (MFT)*

9. How are these properties implemented in Hannotaatio?

10. What new features would you like to see in Hannotaatio (in order to make it even better communication tool?

11. What do you think about the following new feature ideas?

    (a) Ability to record the feedback as a screen capture video instead of static picture?

    (b) Ability to record the screen capture and talk on it?

    (c) Ability to record the screen capture, talk on it and have a webcam recording of the face of the feedback sender?

12. Anything else that you would like to add regarding feedback communication in general or the Hannotaatio tool?

B A4 - What the software is? What it isn't? Why are we building this?

The customer requested an A4 that clearly explains what the software under development really is. This document will be reviewed and updated in the end of each sprint.

"Most important thing that the tool achieves is that it improves customer satisfaction by giving them a chance to participate and communicate with the project team"

- Hanno Nevanlinna

What the software is?

- Website annotation tool
- Extremely usable communication tool between customer and development team
- Tool for customer to report:
    - Design issues
    - Bugs
    - New ideas
- Tool for development team to:
    - Make suggestions to customer
    - Note bugs and link them to bug tracker

What it isn't?

- Drawing tool
- Bug tracking tool (in some way the application can be bug reporting tool, but not a bug tracker)
- Project progress tracking tool
- Discussion board

Why are we building this?

- To improve customer satisfaction
- To make the communication between the customer and the developer team more efficient
- Helping to initialize communication, not to provide the means for a full-length conversation

C Requirements listing

ID	Description	Schedule	Importance for customer	Impact to architecture	Development effort	Status	Related stories (Pivotal Tracker Id + Story name)	Related test cases
FR01	User can initialize the application by clicking a "Annotate" button that redirects the user to the annotation editing site	I1	High	Medium	High	Accepted	5827583 Create the .js file that is going to be added to the website under development 5827718 Capture the webpage DOM tree 5827754 Strip the <script> tags from the captured DOM tree 5828156 Save the captured DOM tree to the backend server 5827768 Capture the styles and send them to backend 5827586 Create a button that captures the DOM and redirects to the editing site 6601141 Generate UUID on the client side 5829468 Send the DOM tree to the backend using normal Ajax POST request 6658749 Decide URLs for capture_tool.js and edit_tool.js 6614775 Modify the CSS style rules with "url" to point to the captured files 5830533 Modify the DOM elements with "src" attribute to point to the captured files	T01
FR02	User can see the Hannotation toolbar and can start annotating the site	I1	High	Medium	High	Accepted	5861079 Create the .js file that is going to be added to the Hannotation editing website 5828106 Create an editing Hannotation site base showing the DOM capture 5827590 Create the Hannotation toolbar	T01
FR03	User can add text on top of the webpage	I1	High	Low	Low	Accepted	5827700 Adding text comments on the webpage from the toolbar	T02
FR04	User can publish his annotation and get a public URL to a screenshot of what he did	I1	High	Medium	High	Accepted	5827909 Make a Publish-button to the Hannotation toolbar 5861697 Save user drawings to the database 5827931 Send user drawings to back-end 5828077 Show user the URL to his published annotation	T03
FR05	User can view a published Hannotation	I1	High	Medium	Medium	Accepted	5861998 Create a viewing mode	T04
FR06	Futurice's development team can install the Hannotation on their website under development	I1	High	High	Medium	Accepted	5828262 Create the JS template that will be copypasted to projects' websites	T05
~~FR07~~	~~User can stop editing the Hannotation (and continue browsing original page) anytime~~	~~I2~~	~~High~~	~~Low~~	~~Low~~	~~Approved~~	~~Investigate possibility of opening the editor in a new tab~~ ~~Implement the back button OR opening the editor in new tab~~	~~T06~~
FR08	User can see when the Hannotation was made	I2	High	Low	Low	Accepted	Show the time of capturing in the editor's view mode	T07
FR09	User can highlight an area on the webpage with a rectangular	I2	Medium	Low	Low	Accepted	5827676 Take Raphael JS into use Implement the drawing of rectangulars	T08
FR10	User can draw a arrow on top of the webpage	I2	Medium	Low	Low	Accepted	Implement the drawing of arrows	T09
FR11	User can remove his drawings	I2	Medium	High	High	Accepted	Implement the selection of a single drawing User can delete a drawing	T10
FR12	User is able to delete a published Hannotation	I2	Medium	Low	Low	Accepted	Create a "delete Hannotation" button to the view mode and send delete command to REST API	T11
FR13	User can see with which browser the published Hannotation was made	I2	Medium	Low	Low	Accepted	Capture the browser when capturing the DOM tree and send it to backend Save/load the browser data through the REST API Show the browser the capturing was made with in the editor's view mode	T07
~~FR14~~	~~"Do you want to save?"~~	~~I2~~	~~Medium~~			~~Approved~~		
FR15	User can edit his drawings	I2	Low	High	High	Accepted	Implement arrow/rectangle resize handles Implement text resize handle Implement moving of drawings by dragging	T12 T13 T14
FR16	User can draw freely on top of the webpage		Low	Medium	Medium	Approved		

ID	Description	Schedule	Importance for customer	Impact to architecture	Development effort	Status	Related stories (Pivotal Tracker Id + Story name)	Related test cases
FR17	Futurice developers can add email addresses that will always get notification when a new Hannotation is published		Low	Medium	Medium	Approved		
FR18	User has to be able to know whether the notifications are sent after publishing the Hannotation		Low	Medium	Medium	Approved	5828099 Collect email addresses where notifications are sent 5861871 Save the collected email addresses to the database 5828168 Send email notifications	
FR19	User can make a new Hannotation based on previously made Hannotation		Low	Medium	High	Approved		
QR01	The DOM tree capturing captures also images from the site	I2	High	High	High	Approved		
QR02	The published Hannotation URL has to be impossible to guess/hack	I1	High	Low	Medium	Accepted		
QR03	A first time user is able to make the annotations without any instructions	I2	High	Low	High	Accepted		
QR04	Saving the Hannotation should not last more than few seconds	I2	High	Medium	Medium	Accepted		
QR05	User is able to use the service from everywhere	I2	High	Low	Low	Accepted		
QR06	The system should be fun to use!	I2	High	Medium	High	Accepted		
QR07	The application should be polished so that it can be taken into production use	I2	High	Low	High	Accepted		
QR08	Loading the Hannotation should not last more than few seconds	I2	Medium	Medium	Medium	Accepted		
QR09	The code has to be maintainable enough so that a developer outside the team can understand the structure and fix bug in a reasonable time	I2	Medium	High	Medium	Approved		
QR10	Futurice's developer should be able to install the Hannotation on their website in less than half an hour	I2	Medium	High	Medium	Accepted	5828262 Create the JS template that will be copypasted to projects' websites	
QR11	The application can be taken into use without setting proxy server configurations	I2	Medium	Medium	Medium	Accepted	Send captured DOM/images/stylesheets/etc to backend using an html form Change the REST API so that it can receive key/value paired data from the html form	
QR12	The system has to be scalable in the cloud server environment	I2	Low	High	Medium	Approved		
QR13	The system should be easy to backup	I2	Low	Low	Low	Approved		
QR14	The system should be portable so that it can be installed to a new environment in less than four hours	I2	Low	High	Medium	Approved		
C01	The system has to work with all common browsers (IE, Firefox, Safari, Chrome), excluding IE6	I2	High	High	High	Approved		
C02	The system has to work without need to install any browser plugins or other software (excluding Flash Player)	I1	High	High	Medium	Accepted		

ID	Description	Schedule	Importance for customer	Impact to architecture	Development effort	Status	Related stories (Pivotal Tracker Id + Story name)	Related test cases
C03	The system should use HTTPS protocol with valid certificate for viewing published Hannotation	I2	High	Low	Medium	Accepted	Configure the server to use HTTPS Use a proper SSL certificate for HTTPS	
C04	The system has to work on iPad	I2	Low	High	High	Approved		
C05	The system has to use only components that allow open sourcing the application in the future	I2	Low	Medium	Medium	Accepted		